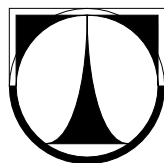


TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií



BAKALÁŘSKÁ PRÁCE

Liberec 2012

Jan Beran

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Řízení tepelné soustavy s dopravním zpožděním pomocí PLC

Control of heat system with transfer delay via PLC

Bakalářská práce

Autor:	Jan Beran
Vedoucí práce:	Ing. Petr Školník, Ph.D
Konzultant:	Ing. Petr Školník, Ph.D

V Liberci 1. 2. 2012

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

V Liberci dne

.....
Jan Beran

Poděkování

Rád bych poděkoval vedoucímu své bakalářské práce Ing. Petru Školníkovi, Ph.D. za podnětné a hodnotné rady, čas strávený při konzultacích a odborné vedení během mé práce. Můj veliký dík patří také přátelům a spolužákům za jejich pomoc, připomínky a náměty.

Abstrakt

Cílem této bakalářské práce je seznámení se s PLC AMiNi4DS od firmy Amit a pomocí něj realizovat řízení tepelné soustavy a poté aplikovat řízení pomocí časových plánů. Předmětem zprávy je úvod do základní hardwarové znalosti a pokročilejšího programování PLC. Tato práce se konkrétně zabývá tvorbou programu a vizualizace pro řízení tepelné soustavy a následně jejich aplikací na reálný model.

V úvodu je zmíněn hardwarový popis a technické parametry PLC AMiNi4DS, které bylo k dispozici pro řešení této úlohy, a také popis modelu tepelné soustavy, který byl řízen.

Dále je popsána realizace propojení PLC a modelu soustavy. V této části zprávy jsou také zmíněny základy programování, jako například programování vstupů a výstupů, a také doplňkové funkce, které byly využity k řešení, nebo mají praktické využití.

Další část zprávy se zabývá identifikací modelu a nalezením vhodných pracovních podmínek. Na základě provedené identifikace jsou zde zpracovány návrhy na možnosti řízení pomocí PI(D) regulátoru.

Závěr zprávy se věnuje již samotnému programovému řešení a tvorbě vizualizace, pomocí které lze model ovládat. Zároveň obsahuje i vyhodnocení týdenního cyklu realizovaného pomocí časových plánů.

Klíčová slova:

AMiNi4DS, regulace, PI(D) regulátor, Smithův prediktor, vizualizace, DetStudio, ViewDet, Matlab, Simulink, simulační schéma, tepelná soustava, identifikace

Abstract

The aim of this Bachelor thesis is familiarity with the PLC AMiNi4DS from Amit, and use it to implement the management of thermal system and then apply the procedure using the schedules. The subject of the message is an introduction to basic hardware knowledge and enhanced the programming of the PLC. This work specifically deals with the creation of the programme and visualization for the management of thermal system and, subsequently, their applications on real model.

In the introduction is mentioned a hardware description and technical parameters of the PLC AMiNi4DS, which was available for the solution of this task, and also a description of the model of a thermal system, which was managed by.

It is further described implementation of the link PLC and the model of the system. In this part of the report also mentioned the foundations of programming, such as the programming of inputs and outputs, and also additional functions, which have been used to deal with, or have a practical use.

The next part of the report deals with the identification of the model, and by finding the appropriate working conditions. On the basis of the carried out identification are processed the proposals to use the PI(D) controller.

The conclusion of the report is dedicated to the software itself no longer a solution and create a visualization that can be used to control the model. At the same time also includes the evaluation of the weekly cycle of using time plans.

Keywords:

AMiNi4DS, control, PI(D) controller, Smith's Predictor, visualization, DetStudio, ViewDet, Matlab, Simulink, simulation scheme, heat system, identification

Obsah

Prohlášení	4
Poděkování.....	5
Abstrakt	6
Klíčová slova:.....	6
Obsah	8
Seznam Obrázků	10
Seznam tabulek	11
Seznam grafů	11
Úvod	12
1. PLC AMiNi4DS	13
1.1 Základní technické parametry.....	14
1.2 Rozmístění konektorů a svorek.....	15
2. Model tepelné soustavy	16
3. Propojení PLC AMiNi4DS s modelem tepelné soustavy	17
4. Představení a úvod do programování v DetStudios	18
4.1 Základní nastavení projektu.....	18
4.2 Databáze	18
4.3 Procesy.....	19
4.4 Programování vstupů a výstupů.....	20
4.4.1 Číslicové vstupy	20
4.4.2 Číslicové výstupy	20
4.4.3 Analogové vstupy.....	20
4.4.4 Analogové výstupy.....	21
4.5 Rozšiřující funkce.....	21
4.6 Programování obrazovek.....	22
5. ViewDet a tvorba vizualizace	23
5.1 Zakládání proměnných a aliasů.....	23
5.2 Scéna.....	23
6. Identifikace	25
6.1 Tvorba základního programu.....	25
6.2 Měření přechodové charakteristiky	27
6.3 Zpracování přechodové charakteristiky v programu Matlab	29
7. Návrh regulátoru.....	33
7.1 Návrh PID regulátoru.....	33
7.2 Návrh Smithova prediktoru.....	38
8. Programové řešení regulátoru	39
8.1 PID regulátor.....	39
8.2 Smithův prediktor.....	41
9. Vyhodnocení regulace PID regulátoru	43
10. Časové plány	45
11. Kompletní program v DetStudios	49
12. Vizualizace	49
Závěr	53

Seznam použité literatury.....	55
Příloha I – Katalogový list PLC AMiNi4DS	56
Příloha II – Model tepelné soustavy s popisem.....	58
Příloha III – Skript pro výpočet doby náběhu a průtahu	59
Příloha IV – Smithův prediktor.....	60
Příloha V – Proměnné a Aliasy	61
Příloha VI – Přehled vytvořených obrazovek.....	62
Příloha VII – Jednotlivé scény vizualizace.....	63

Seznam Obrázků

Obrázek 1.1: PLC AMiNi4DS	13
Obrázek 1.2.1: Rozmístění a popis konektorů a svorek	15
Obrázek 2.1: Schéma modelu tepelné soustavy	16
Obrázek 3.1: Ilustrace použitého konektoru typu CANON 25	17
Obrázek 3.2: Štítkové údaje čidla PT100	17
Obrázek 4.2.1: Okno s definicí proměnné	18
Obrázek 6.1.1: Programová část zapisující hodnoty na výstupy	25
Obrázek 6.1.2: Programový část řešící načtení teplot a jejich archivaci	26
Obrázek 6.2.1: Přechodová charakteristika	28
Obrázek 6.3.1: Základní okno funkce ident	29
Obrázek 6.3.2: Okno položky process models	30
Obrázek 6.3.3: Aproximovaný systém prvního řádu s dopravním zpožděním, přesnost 97,03%	31
Obrázek 6.3.4: Aproximovaný systém druhého řádu s dopravním zpožděním, přesnost 97,79%	31
Obrázek 6.3.5: Aproximovaný systém prvního řádu bez dopravního zpoždění, přesnost 95,36%	32
Obrázek 6.3.6: Aproximovaný systém druhého řádu bez dopravního zpoždění, přesnost 95,47%	32
Obrázek 7.1.1: Výpočty parametrů regulátoru	35
Obrázek 7.1.2: Simulační schéma s blokem PID	35
Obrázek 7.1.3: Záložka nastavení Bloku PID	36
Obrázek 7.1.4: Rozhraní s možností úpravy regulačního průběhu	36
Obrázek 7.1.5: Porovnání regulačních průběhů systému prvního řádu	37
Obrázek 7.1.6: Porovnání regulačních průběhů systému druhého řádu	37
Obrázek 7.2.1: Schéma Smithova prediktoru	38
Obrázek 8.1.1: Simulační schéma pro porovnání výpočtů akčního zásahu	40
Obrázek 8.1.2: Porovnání regulačních průběhů	40
Obrázek 8.1.3: Programové řešení PID regulátoru	41
Obrázek 8.2.1: Simulační schéma Smithova prediktoru s náhradním obvodem PID regulátoru a s číslicovým výpočtem PID regulátoru	41
Obrázek 8.2. 2: Porovnání regulačních průběhů Smithova prediktoru	42
Obrázek 10.1: Nastavení hodnot časového plánu pro ventilátor č. 2	47
Obrázek 10.2: Průběh teploty y4 při měření časových plánů	48
Obrázek 10.3: Průběh teploty y4 při měření víkendového režimu	48
Obrázek 12.1: Nastavení pozadí scény	49
Obrázek 12.2: Nastavení prvku Proměnná	50
Obrázek 12.3: Nastavení prvku Text	50
Obrázek 12.4: Nastavení prvku PCArchiv	51
Obrázek 12.5: Nastavení prvku Archiv	51
Obrázek 12.6: Nastavení proměnných prvku Časový plán	52
Obrázek 12.7: Nastavení parametrů prvku Časový plán	52

Seznam tabulek

Tabulka 1.1.1: Přehled základních technických parametrů.....	14
Tabulka 3.1: Zapojení pinů a konektoru svorkovnice (ze strany modelu)	17
Tabulka 7.1.1: Doby náběhu a průtahu pro přenos prvního a druhého řádu	33
Tabulka 7.1.2: Výpočty parametrů regulátoru	33
Tabulka 7.1.3: Parametry regulátoru.....	33
Tabulka 7.1.4: Výpočty parametrů regulátoru	34
Tabulka 7.1.5: Parametry regulátoru.....	34
Tabulka 7.1.6: Parametry regulátoru.....	35
Tabulka 7.2.1: Parametry Smithova prediktoru	38
Tabulka 10. 1: Matice časových zlomů.....	46
Tabulka 10.2: Matice hodnot pro časové zlomy	46

Seznam grafů

Graf 6.2.1: Porovnání charakteristik při různém výkonu čerpadla	27
Graf 9.1: Průběh regulace teploty y4 modelu soustavy	43
Graf 9.2: Odezva na spuštěnou chybu.....	44

Úvod

Úkolem této bakalářské práce je řídit za pomoci PLC model tepelné soustavy, který představuje místnost nějakého objektu, nebo objekt samotný. Tato varianta, kdy se řídí především vytápění objektu pomocí PLC, je nejen velice efektivní ale zároveň velmi využívána v tzv. inteligentních domech. Ty jsou velmi rozšířené a navíc nabízí možnost řízení téměř celého domu, od vytápění až po ovládání jednotlivé elektroniky. Tento trend se stal za poslední roky opravdu populární a díky svým možnostem rok od roku více využívaný.

Pro řešení bakalářské práce bylo k dispozici PLC české firmy Amit, AMiNi4DS. Model, který se jím řídil, obsahoval z pohledu řízení a regulace dva hlavní prvky. Průtokový ohřívač, představující kotel, a tepelný výměník voda – vzduch, který simuloval domácí radiátor.

V úvodu práce představím PLC AMiNi4DS. Především jeho hardwarové řešení a technické parametry. Dále zde uvedu i popis modelu tepelné soustavy, která byla řízena.

Další kapitola bude věnována popisu propojení PLC s modelem tepelné soustavy.

V následující kapitole se budu věnovat identifikaci parametrického modelu tepelné soustavy, tedy z jakých dat a jak byla provedena a jejímu výsledku.

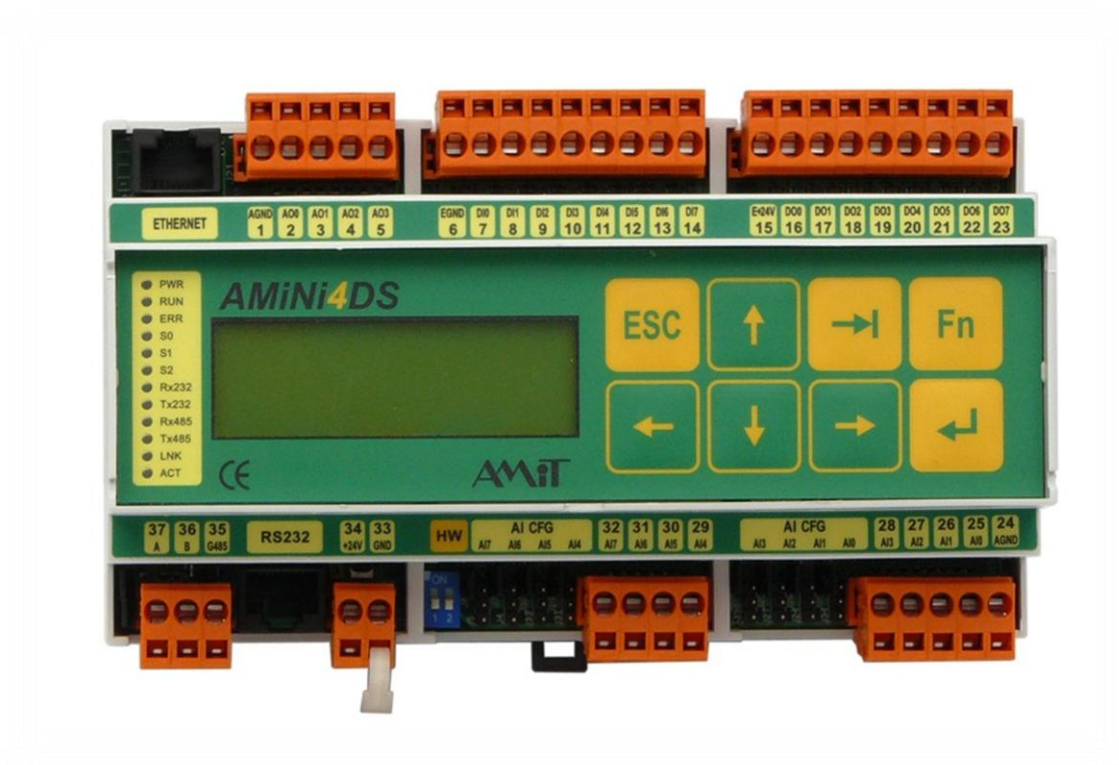
Na základě získaného parametrického modelu se budu zabývat návrhem vhodného PI(D) regulátoru a jako další možnost řízení představím Smithův prediktor, který se využívá u soustav s větším dopravním zpožděním.

Dále se budu zabývat tvorbou programu a vizualizace. Program se bude skládat z několika částí, z čehož ty hlavní jsou ovládání a zpracování vstupů a výstupů a samotná regulace. Co se týče vizualizace, úkolem zde bylo vytvořit pro uživatele přehledné a snadné ovládání a sledování jednotlivých stavů.

V závěru práce se budu věnovat vyhodnocení kvality regulace a také vyhodnocení nepřetržitého týdenního cyklu řízení, během kterého byly vyzkoušeny různé režimy řízení. K tomu bylo využito časových plánů.

1. PLC AMiNi4DS

AMiNi4DS je volně programovatelný automat, který je vhodný pro komplexní autonomní řízení a ovládání malých soustav, strojů či zařízení. Především se používá na rozsáhlé distribuované systémy měření a regulace, řízení jednoduchých strojů a zařízení, automatizaci budov, inteligentních domů a na monitoring a archivaci měřených dat.



Obrázek 1.1: PLC AMiNi4DS

1.1 Základní technické parametry

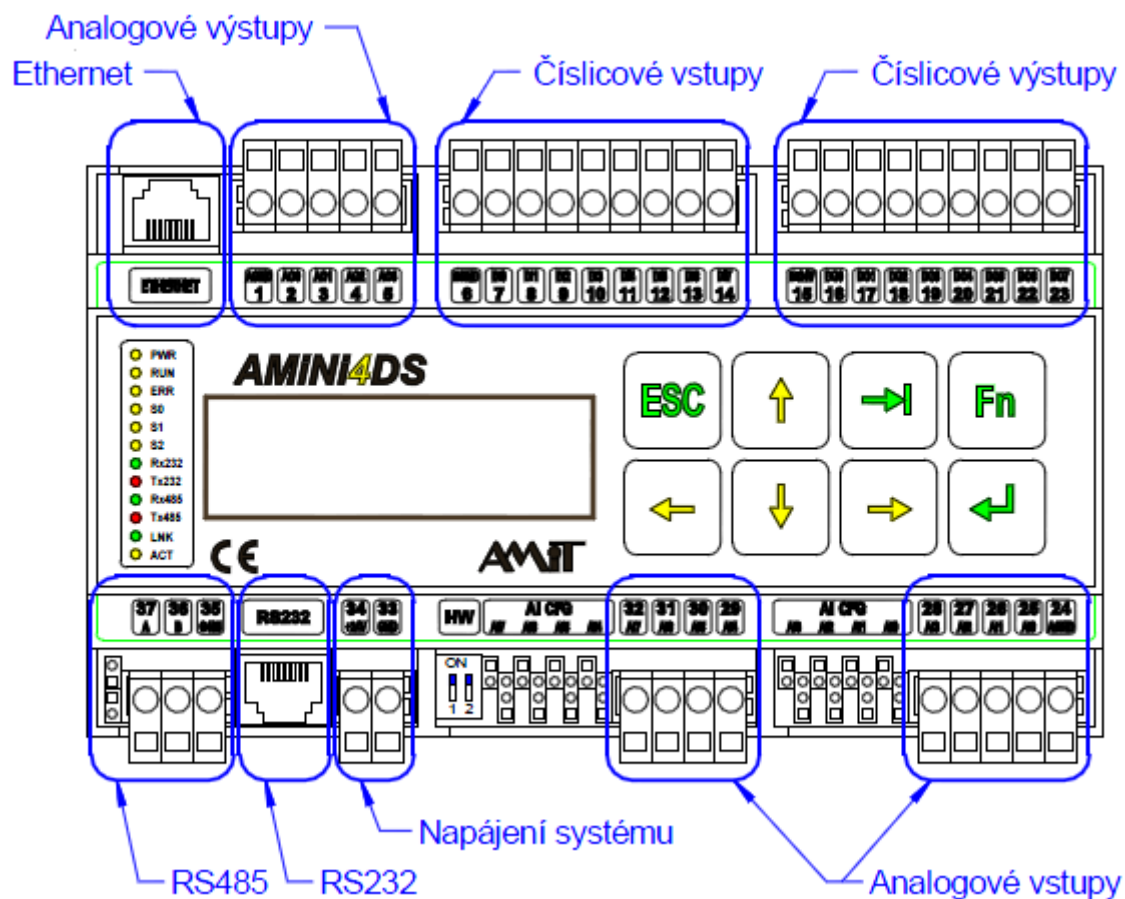
V Tabulce 1.1.1 je uveden jen základní přehled technických parametrů, které jsou důležité pro běžné uživatele. Kompletní katalogový list je uveden v příloze I.

Tabulka 1.1.1: Přehled základních technických parametrů

CPU	ST10F269
Paměť Flash	256 + 1024 KB
Paměť RAM (zálohovaná)	1024 KB
EEPROM	2 KB
Zálohování RAM	Lithiová baterie Panasonic, 5 let
Vstupy	8 × Digitální + 8 × Analogové
Číslicové vstupy	24 V ss./stř.
Analogové vstupy	0 .. 5 V / 0 .. 10 V / 0 .. 20 mA / Ni1000 / Pt1000
Výstupy	8 × Digitální + 4 × Analogové
Číslicové výstupy	24 V/0,3 A ss.
Analogové výstupy	0 .. 10 V (max. 10 mA)
Komunikace	
Sériový komunikační kanál	1 × RS232 (RJ45), dle EIA-561
	1 × RS485 (Konektory WAGO)
Ethernet	1 × 10 Mbps, RJ45, dle IEEE802.3
Napájení	24 V ss. ±20 %
Odběr (bez výstupů)	Max. 200 mA při 24 V ss.
Ostatní	
Displej	Grafický, 122 × 32 bodů
Klávesnice	8 tlačítek
Pracovní teplota	0 .. 50 °C
Maximální vlhkost okolí	< 95 % nekondenzující
Montáž	DIN lišta 35 mm
Hmotnost	500 g
Rozměry (š × v × h)	160 × 98 × 74 mm

1.2 Rozmístění konektorů a svorek

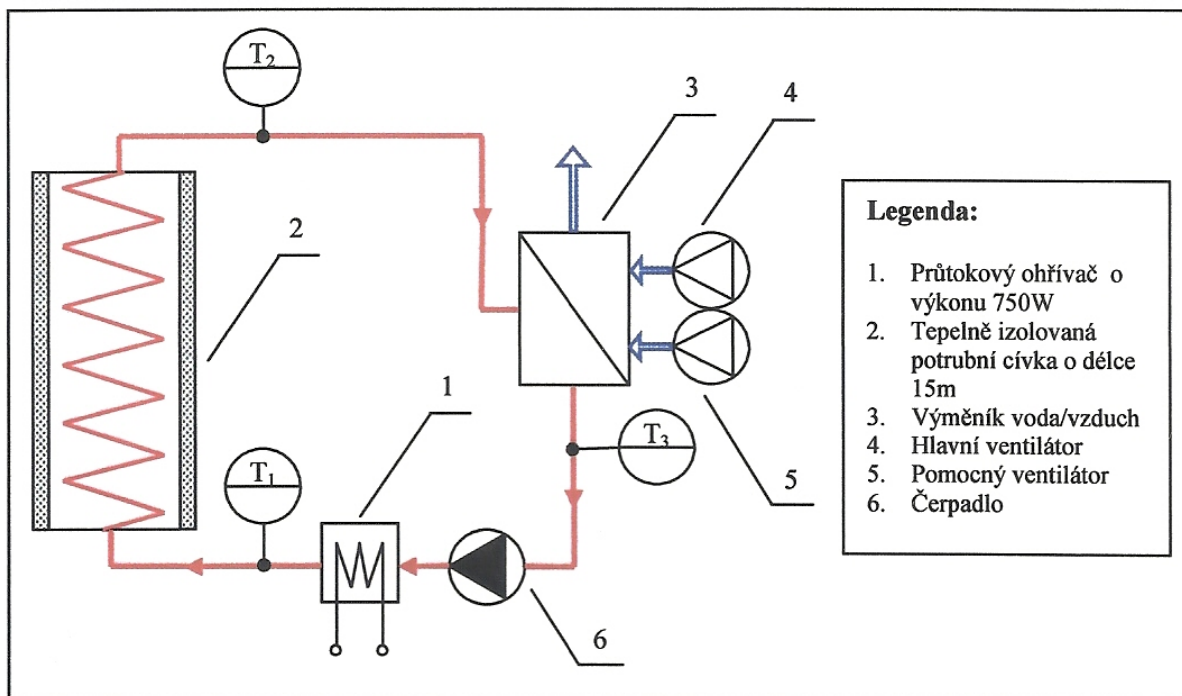
Propojkami, které jsou u analogových vstupů, se nastavuje typ vstupu. Vstupy je možné nezávisle konfigurovat pro rozsahy 0 - 5 V, 0 - 10 V a 0 - 20 mA nebo pro přímé připojení čidel Ni1000 / Pt1000.



Obrázek 1.2.1: Rozmístění a popis konektorů a svorek

2. Model tepelné soustavy

Model tepelné soustavy je založen na principu přenosu tepla pomocí teplotonosného média prostřednictvím potrubního systému. Reálné vyobrazení je uvedeno v příloze II. Blokové schéma modelu tepelné soustavy je uvedeno na Obrázku 2.1.



Obrázek 2.1: Schéma modelu tepelné soustavy

Teplotonosné médium, v tomto případě voda, je transportováno pomocí spojitě regulovatelného čerpadla (6) do průtokového ohřívače (1) o výkonu 750 W. Teplota vody vystupující z ohřívače je měřena platinovým teploměrem (Pt1000) T1. Ohřátá voda dále vstupuje do tepelně izolované měděné potrubní cívky (2) dlouhé 15m. Zde vzniká v závislosti na zvolených otáčkách čerpadla dopravní zpoždění v rozmezí 50 – 200s. Spotřebič tepelné energie představuje výměník tepla typu voda/vzduch (3), který předává tepelnou energii do okolního prostředí. Teplota vzduchu vystupující z výměníku je měřena teploměrem Pt100, kterým byla úloha rozšířena, a proto není zobrazen ve výše uvedeném schématu. Úroveň spotřeby tepla lze nastavit pomocí dvou regulovatelných ventilátorů (4, 5). Ventilátor 1 (5) lze ovládat pouze dvoustavově (zapnuto / vypnuto), ventilátor 2 (4) je řízen spojitě. Teplota vody vstupující do výměníku a vystupující z výměníku je měřena teploměry T2 a T3. Z výměníku se voda vrací zpět do čerpadla a celý koloběh se opakuje.

Model tepelné soustavy je potřeba ovládat 2 digitálními vstupy, kterými je ovládán průtokový ohřívač (1) a ventilátor (5). Dále jsou potřeba 2 analogové vstupy na ovládání ventilátoru (4) a čerpadla (6). Potřeba jsou také 4 analogové výstupy pro sledování naměřené teploty z teploměrů T1, T2, T3 a T4 (Pt100).

3. Propojení PLC AMiNi4DS s modelem tepelné soustavy

Protože model tepelné soustavy má na výstupu konektor typu CANON 25 a u AMiNi4DS se vstupy a výstupy propojují pomocí propojovacích kabelů, bylo propojení realizováno pomocí vyrobeného kabelu. K jeho realizaci byl zakoupen protikus konektoru CANON 25 (viz. Obrázek 3.1) a 34 žilový IDE kabel, který byl upraven na 25 žilový. Do PLC byly zapojeny volné konce kabelu, jejich popis je uveden v níže uveden tabulce (Tabulka 3.1).

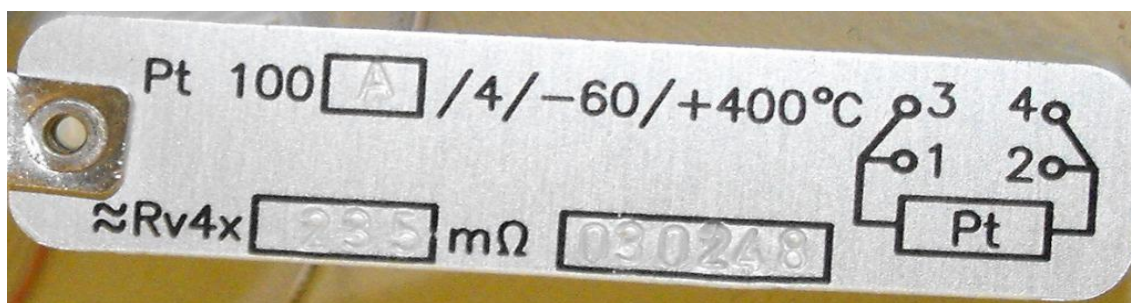


Obrázek 3.1: Ilustrace použitého konektoru typu CANON 25

Tabulka 3.1: Zapojení pinů a konektoru svorkovnice (ze strany modelu)

Pin	Signál konektoru	Směr	Napěťová úroveň
11	Měřená veličina y1	výstup	0 - 10V
12	Měřená veličina y2	výstup	0 - 10V
13	Měřená veličina y3	výstup	0 - 10V
17	GND	----	0V
19	Ventilátor 2	vstup	0 - 10V
20	Čerpadlo	vstup	0 - 10V
23	Ventilátor 1	vstup	TTL
24	Topení	vstup	TTL

Čidlo PT100, jímž se model soustavy rozšiřoval, se běžně připojuje k PLC AMiNi4DS přímo na analogové vstupy, které mají integrovaný napěťový dělič. Ten ale nebylo možné použít kvůli nízkému odporu, a tím i nízkému rozlišení, které bylo cca 4 °C. Proto bylo čidlo připojeno pomocí diferenčního (rozdílového) zesilovače, čímž se rozsah čidla zvýšil na rozlišení v rozmezí 0 – 100 °C, čemuž odpovídal napěťový rozsah 0,6 – 6,9V.



Obrázek 3.2: Štítkové údaje čidla PT100

4. Představení a úvod do programování v DetStudiu

DetStudio je návrhové prostředí, které je určeno pro tvorbu uživatelských aplikací pro všechny standardní řídicí systémy firmy AMiT. V tomto vývojovém prostředí lze vytvořit vlastní aplikaci, navrhnout a odsimulovat vzhled obrazovek zobrazovačů řídicích systémů, definovat chybová hlášení, on-line ladit běžící aplikaci nebo i vytvořit dokumentaci vytvořeného programu ve formě HTML.

4.1 Základní nastavení projektu

Jedna z prvních položek, které je potřeba nastavit před začátkem tvorby programu, je typ stanice. Dále je zde možné vyplnit údaje o autorovi, projektu nebo stručný popis.

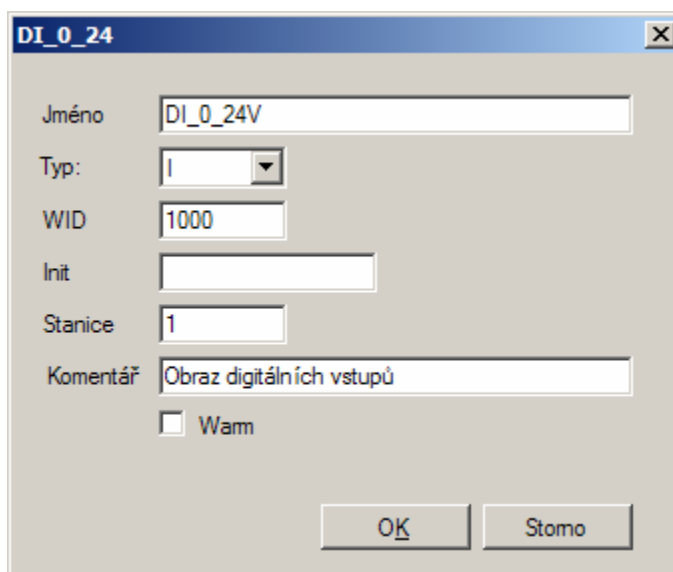
V další položce se nastavuje tzv. ID1 a ID2. To jsou řetězce, které řídicí systém „vrátí“ po dotazu na identifikaci, která vždy probíhá nahráním programu do řídicího systému. Takto lze jednoduše zjistit, jestli je řídicí systém správně připojen.

Jako další je potřeba, zda se řídicí systém bude chovat jako pasivní nebo aktivní stanice. V případě pasivní stanice řídicí systém nekomunikuje s ostatními systémy. Pokud bude jako aktivní stanice, tak může komunikovat s ostatními v systému na principu MASTER/SLAVE.

Jako poslední položku, kterou je potřeba nastavit, je komunikace. Zde se musí nastavit komunikační port, obvykle COM, adresa stanice (1 - 31), způsob komunikace (RS232, ethernet...) a komunikační rychlost (9600 – 115200 Bd).

4.2 Databáze

Databáze se skládá z proměnných a aliasů. Při zakládání proměnných, je potřeba nastavit její jméno, typ, WID a popřípadě popis a inicializační hodnotu (viz Obr. 4.2.1). Jméno proměnné se píše bez diakritiky, tedy bez háčků a čárek, a mezery se nahrazují podtržítkem.



Obrázek 4.2.1: Okno s definicí proměnné

Typem proměnné se myslí její datový typ. V DetStudiosu se využívá Integer (I), Long (L) a Float (F), MI[x,y] (matice typu I), ML[x,y] (matice typu L) a MF[x,y] (matice typu F), kde x a y jsou rozměry matice. Důležitý je WID, číselný indikátor v rozsahu 0 – 65500. Toto číslo je používáno při přístupu k proměnné a musí být jedinečné v celé aplikaci (tedy i v síti řídicích systémů). Přidělování WID řeší DetStudio automaticky a nedoporučuje se jej editovat (až na výjimečné případy). Každý WID je sestaven z čísla řídicího systému na síti DB-Net a pořadového čísla proměnné v konkrétním řídicím systému. Tak je zajištěno, aby nedocházelo ke kolizím identifikátoru WID v aplikaci. Do pole Init lze zadat hodnotu proměnné, kterou bude nabývat při spuštění programu.

U celočíselných proměnných typu I (MI) / L (ML) je možnost přistupovat k nim i jako k bitovým proměnným. Každá může nabývat 16 / 32 hodnot TRUE / FALSE. Bity jsou pak identifikovány jménem proměnné a číslem bitu (jméno.číslo). Jednotlivým bitům lze pak také přiřadit jméno a poté se na ně přímo odkazovat. K tomu slouží tzv. Alias. V programu se pozná podle znaku @, který má před svým jménem. Tento Alias pak nahrazuje odkaz pomocí jména proměnné a čísla bitu:

@Alias = jméno.číslo

4.3 Procesy

Činnost řídicího systému probíhá sekvenčně a je rozdělena do tzv. procesů. Každý proces obsahuje část programu, která může pracovat samostatně a nezávisle na ostatních procesech. Jsou tři typy procesů:

- Proces LA – pracuje s vrcholem zásobníku
- Proces RS – pracuje s reléovými schémata
- Proces ST – pracuje s klasickým strukturovaným textem

Velikou výhodou je, že jak proces ST nebo RS, má již předdefinované moduly a reléová schémata, která stačí jen do programu vložit a doplnit je o proměnné, aliasy nebo jen o parametry, takže zde odpadá problém s tím, abychom si některé složité bloky programu museli sami programovat.

Procesy lze také dělit podle priority vykonávání a lze jim nastavit periodu vykonávání:

- *Interrupt* (Interrupt_0 - 15) – mají nejvyšší prioritu
- *Velmi rychlé procesy* (ProcHi_0 -1) – umí vyvolat přerušení ostatních procesů s nižší prioritou (perioda 1 – 1677 ms)
- *Rychlý proces* (Quick) - umí vyvolat přerušení ostatních procesů s nižší prioritou (perioda 5, 10, 20, 50 a 100 ms)
- *Řádné procesy* (Normal_0 - 15) – neumí vyvolat přerušení ostatních procesů (perioda 0,1 – 1000000 s)
- *Prázdný proces* (Idle) – spuštěn v okamžiku, kdy si žádný ze zbývajících procesů nenárokuje čas procesu
- *Proces vykonávaný pouze jednou* (Init) – vykoná se jako první po teplém/studeném startu

4.4 Programování vstupů a výstupů

4.4.1 Číslicové vstupy

Řídicí systém AMiNi4DS má 8 číslicových vstupů, které se značí DI.0 – DI.7, které lze použít pro stejnosměrný i střídavý signál. Vyhodnocení záleží na programu. Načíst hodnotu ze vstupů lze provést dvěma způsoby. Můžeme načíst všech 8 číslicových vstupů najednou:

```
DigIn #0,DigVstup,0x0000
```

kde:

DigIn – modul načítání číslicových vstupů

#0 – číslo logického kanálu

DigVstup – proměnná, do které se hodnota bude zapisovat

0x0000 – příznak negace

Nebo lze načíst pouze 1 číslicový vstup, tedy 1 bit:

```
BinIn #0.1,0,VstupX.1
```

kde:

BinIn – modul načítání 1 číslicového vstupu

#0.1 – číslo bitu logického kanálu

0 – negace bitu (1 = negace)

VstupX.1 – proměnná, nebo alias do kterého se hodnota zapisuje

4.4.2 Číslicové výstupy

Číslicových výstupů je zde opět 8 a značí se DO.0 – DO.7. Možnosti programového řešení jsou stejné jako u číslicových vstupů. Rozdíl je v použitém modulu. Jen se nahradí *DigIn* *DigOut* a *BinIn* *BinOut*:

```
DigOut #0,DigVystup,0x0000
```

```
BinOut #0.1,0,VystupX.1
```

Důležitou poznámkou, kterou uvádí výrobce, je, že pro správnou funkci číslicových vstupů a výstupů je potřeba připojit svorky E+24V a EGND na zdroj napětí. Důležitou poznámkou, kterou ale již výrobce neuvádí je, že napětí, které se připojí na tyto svorky, je i výstupní napětí na číslicových výstupech, a proto jejich označení je částečně matoucí. Není tedy potřeba připojovat 24V, ale připojí se takové napětí, které chceme mít na číslicovém výstupu.

4.4.3 Analogové vstupy

Řídicí systém AMiNi4DS má 8 analogových vstupů, které se značí AI.0 – AI.7. V DetStudiosu jsou jim implicitně přiřazeny názvy AI00_0 – AI00_7. Pomocí konfiguračních propojek, které jsou umístěny hned u analogových vstupů, lze daný analogový vstup konfigurovat pro rozsahy 0 - 5 V, 0 - 10 V, 0 - 20 mA, nebo pro přímé

připojení čidla Ni1000, Pt1000. Pokud nemáme přímo připojené čidlo, tak se analogové vstupy načítají pomocí modulu *AnIn*:

```
AnIn    #AI00_0, Value, Rozsah, Elmin, Elmax, Fyzmin, Fyzmax
```

kde:

AnIn – modul načítání analogového vstupu

#AI00_0 – číslo analogového vstupu

Value – proměnná, do které se hodnota zapisuje

Rozsah – rozsah elektrické veličiny (pro 0 – 5V je *Rozsah* = 5)

Elmin – minimální hodnota elektrické veličiny

Elmax – maximální hodnota elektrické veličiny

Fyzmin – minimální hodnota fyzikální veličiny, které je minimální hodnota elektrické veličiny úměrná

Fyzmax – maximální hodnota fyzikální veličiny, které je maximální hodnota elektrické veličiny úměrná

Pokud máme přímo připojené čidlo typu Ni1000 nebo Pt1000, použijeme modul *Ni1000*:

```
Ni1000  #AI00_0, TEPLOTA, 6180
```

kde:

Ni1000 – modul načítání hodnoty analogového vstupu

TEPLOTA – proměnná, do které se hodnota zapisuje

6180 – citlivost snímače [ppm] (5000 / 6180)

4.4.4 Analogové výstupy

Analogové výstupy jsou pouze 4, jejichž výstupní rozsah napětí je 0 – 10V. Značí se AO.0 – AO.3. Programují se stejně jako analogové vstupy, jen se místo modulu *AnIn* použije modu *AnOut*:

```
AnOut    #AO00_0, Value, Rozsah, Elmin, Elmax, Fyzmin, Fyzmax
```

4.5 Rozšiřující funkce

Mezi nejdůležitější rozšiřující funkce můžeme zařadit programovou archivaci hodnot, provozní deník, správce dat nebo správce archivů.

Pomocí archivace hodnot můžeme přímo definovat archívy, které poté můžeme vyčítat, ukládat a následně zpracovávat v počítači. Archívy se ukládají v zálohované paměti RAM řídicího systému. Množství ukládaných dat je dáno především velikostí

volného místa v paměti řídicího systému. V případě většího množství volného místa v paměti lze ukládat i deseti tisíce hodnot. Pro práci s archívy slouží moduly *SyncArch* a *SyncMark*, přičemž modul *SyncMark* se využívá ke generování tzv. časových značek pro modul *SyncArch*, který ukládá nastavené hodnoty do předem definovaných matic. Časovou značku si můžeme libovolně nadefinovat na základě časové jednotky (sekundy, hodiny až měsíce) a času, kdy se archivace provede. Například lze nastavit synchronizace na každou hodinu, nebo na každý třetí den o libovolném čase.

Další užitečnou funkcí je provozní deník. Ten obsahuje chybová a jiná hlášení, které se týkají samotného provozu řídicího systému. Provozní deník je dvojího typu. Systémový, který je možné prohlížet pouze na terminálu, který je připojen k řídicímu systému a má hloubku 50 - ti hlášení. Druhý je aplikační, který je přístupný přes počítač a je možné ho zpracovávat i jinými programy. Jeho hloubka je omezena pouze velikostí volné paměti řídicího systému. K vytvoření je potřeba založit 2 proměnné o speciálním WIDu xx900 a xx901, kde xx je číslo stanice. Proměnná o WIDu xx900 musí být typu Integer a louží jako index hlášení. Druhá proměnná o WIDu xx901 je matice typu Integer do které se zapisují hlášení.

Správce dat slouží pro uchování aktuálních hodnot proměnných v řídicím systému. Načtená data se ukládají do samostatných souborů s příponou .psps. Dále je umožněno jejich prohlížení ve formě přehledných výpisů, uložení inicializačních hodnot do otevřeného projektu nebo také možnost porovnání dat ve dvou souborech nebo s aktuálním stavem proměnných v řídicím systému.

Správce archívů slouží pro načítání, prohlížení, export a analýzu dat časových archívů a provozních deníků přímo v prostředí DetStudia. Načtená data se ukládají do souboru s příponou .pspa.

4.6 Programování obrazovek

Další z užitečných možností je programování obrazovek. Lze zde vytvářet obrazovky, které mohou obsahovat pouze informace o aktuálním stavu, obrazovky typu login, menu a mnoho dalších. Pomocí velkého množství funkcí lze i provázat program s ovládáním přímo na řídicím systému a přes něj měnit hodnoty proměnných, spouštět různé bloky programu atd. Designová a základní programová tvorba obrazovek je na principu vkládání a editování jednotlivých prvků. Pokročilejší programování lze provádět pomocí tzv. skriptů.

5. ViewDet a tvorba vizualizace

ViewDet je samostatný servisní nástroj pro Windows. Doplnuje a rozšiřuje možnosti návrhového prostředí DetStudio v oblasti sledování, ladění a nastavení aplikace v řídicím systému. V rámci těchto vlastností se ViewDet využívá k tvorbě jednoduchých vizualizací.

Základní vlastností je možnost čtení a zápisu jednotlivých hodnot databázových proměnných a aliasů. Hlavní předností programu ViewDet je, že umí číst, zobrazovat, tisknout a exportovat archivy a provozní deník z řídicího systému, včetně pamatování si jejich historie. Zobrazované proměnné lze umístit libovolně v tzv. "scéně" s možností vložení vlastního obrázku na pozadí - takto lze vytvořit velmi jednoduchou vizualizaci procesů. ViewDet umožňuje i kompletní editaci IP konfigurace stanic připojených k průmyslovému Ethernetu.

Mezi hlavní výhody aplikace ViewDet bezesporu možnost uskupení zobrazovacích prvků ve scénách a rychlé přepínání mezi nimi, možnost vložení vlastního pozadí ve scéně, zobrazování hodnot formou grafu, tisk scény s prvky a snadný export zarchivovaných dat.

5.1 Zakládání proměnných a aliasů

Stejně jako v DetStudiosu lze proměnné a aliasy zakládat v záložkách „proměnné“ a „aliasy“. Způsob jejich tvorby je totožný jako v DetStudiosu. Pokud se tvoří vizualizace pro již vytvořený program, tak aplikace ViewDet nabízí veliké usnadnění v této oblasti, a to takové, že proměnné a aliasy lze načíst přímo z PLC nebo je importovat ze zdrojového souboru vytvořeného programu.

5.2 Scéna

Každou ze scén si lze představit jako volnou pracovní plochu, na kterou lze vkládat různé, předdefinované, zobrazovací/editační prvky. Při práci s programem je možné mít otevřeno více scén v jednom okamžiku a libovolně se mezi nimi přepínat. Jak již bylo zmíněno, lze do vizualizace vkládat vlastní pozadí, to lze nastavit v parametrech scény při její editaci.

Mezi scénické prvky patří:

- Inspektor – prvek, ve kterém lze sledovat hodnoty většího množství proměnných
- Matice – prvek zobrazující hodnoty matice definované v řídicím systému
- Archiv – prvek zobrazující archiv vzorků uložený v řídicím systému
- PC Archiv – prvek umožňující archivaci načtených vzorků do PC
- Provozní deník – prvek spravující provozní deníky
- Časový plán – prvek umožňující komfortní správu časových plánů programovaných funkčním blokem DayPlan
- Text – textový popis
- Proměnná – prvek zobrazující jednoduché okno obsahující hodnotu definované proměnné

Editace scénických prvků je snadná a intuitivní. Téměř vždy se jenom nastavuje jméno prvku, poté se vybírá proměnná, popřípadě archiv, v jakém formátu se bude proměnná nastavovat, zda-li je proměnná jen ke čtení, nebo jestli ji lze editovat a jako

poslední se nastavuje pozice, kam chceme prvek umístit. U prvku PC Archiv se navíc nastavuje, zda chceme vybrané proměnné zobrazovat formou grafu a jak „hluboký“ by archiv měl být. Zde je možnost vybrat například vybrat den, měsíc nebo neomezený.

6. Identifikace

Identifikace soustavy byla prováděna jako odezva na jednotkový skok, tedy měření teploty nad výměníkem (teplota y4) v závislosti na sepnutém topení. K tomuto měření byl vytvořen základní program na ovládání prvků soustavy a snímání a archivaci hodnot z teplotních čidel.

Před samotným měřením přechodové charakteristiky, bylo potřeba zvolit správné počáteční podmínky. Jejich nastavení spočívalo v nastavení výkonu čerpadla a stálým sepnutím ventilátoru č. 1, který se ovládá diskretně. Model tepelné soustavy byl navržen tak, že sepnutý ventilátor slouží k ustálení odběru tepla, tedy ustálení teploty nad výměníkem.

Ke zpracování naměřené přechodové charakteristiky byl využit program Matlab a jeho identifikační nástroj „ident“.

6.1 Tvorba základního programu

Základní část programu se skládá ze dvou částí. První část programu (Obr. 6.1.1) se věnuje zapisování hodnot na výstupy PLC. K tomu byly použity již zmíněné moduly AnOut, BinOut, DigOut. Navíc je zde přidána možnost ovládat digitální výstupy pomocí pulsně šířkové modulace (dále jen PWM). Hlavními parametry funkce PWM jsou minimální a maximální délka pulsu a vstupní proměnná, ve které je obsažena procentuální hodnota. Na základě hodnoty vstupní proměnné se nastaví délka pulsu, který bude odeslán na digitální výstup.

```
//zápis hodnoty proměnné pro ovládání ventilátoru2 AO_1 na analogový výstup číslo 0, hodnota
//proměnné se zadava v rozsahu 0 - 100%, tato hodnota se posleze převede na uroveň napeti 0 - 10V
//ventilátor 1
AnOut #AO00_1, ventilator, 10.000, 0.000, 10.000, 0.000, 100.000
//zápis hodnoty proměnné pro ovládání čerpadla AO_1 na analogový výstup číslo 1, hodnota
//proměnné se zadava v rozsahu 0 - 100%, tato hodnota se posleze převede na uroveň napeti 0 - 10V
//čerpadlo
AnOut #AO00_0, cerpadlo, 10.000, 0.000, 10.000, 0.000, 100.000
//zápis hodnoty bitu pro ovládání topení na digitalní výstup číslo 1
//topení
BinOut @topeni, 0x0000, #0.0
//ovládání topení pomocí PWM
PWM 0x0009, topeni, perioda_pwm, 0.000, perioda_pwm, @topeni
//zápis hodnoty bitu pro ovládání ventilátoru1 na digitalní výstup číslo 1
//ventilátor 2
BinOut @vent, 0x0000, #0.1
Pwm 0x0009, vent, 10.000, 0.000, 10.000, @vent
//zápis hodnoty proměnné ovladani na všechny digitalní výstupy
//pomocná proměnná pro ovládání topení a ventilátoru 2
DigOut ovladani, #0, 0x0000
```

Obrázek 6.1.1: Programová část zapisující hodnoty na výstupy

Ve druhé části programu (Obr. 6.1.2) načítají hodnoty z teplotních čidel a následně se každých 10 vteřin ukládají do matice arc_tepl. Načítání hodnot probíhá modulů AnIn. Archivace hodnot se skládá ze dvou prvků. Prvním je SynMark, který generuje tzv. časovou značku, která spouští archivaci. Je tedy vstupem druhého modulu, modulu SyncArch. Modulu SyncArch je nutné nastavit několik parametrů. Proměnnou, kterou budeme archivovat, do jaké matice ji budeme ukládat, matici s indexy a matici s casy, kdy archivace proběhla.

```

//načtení analogového vstupu číslo 0, následný převod naměřené
//hodnoty napětí na teplotu a následný zápis do proměnné y1
AnIn #AI00_0, y1, 10.000, 0.000, 10.000, 0.000, 100.000
//načtení analogového vstupu číslo 1, následný převod naměřené
//hodnoty napětí na teplotu a následný zápis do proměnné y2
AnIn #AI00_1, y2, 10.000, 0.000, 10.000, 0.000, 100.000
//načtení analogového vstupu číslo 2, následný převod naměřené
//hodnoty napětí na teplotu a následný zápis do proměnné y3
AnIn #AI00_2, y3, 10.000, 0.000, 10.000, 0.000, 100.000
//načtení analogového vstupu číslo 2, následný převod naměřené
//hodnoty napětí na teplotu a následný zápis do proměnné y4

//archivace teplot
//generuje časovou značku na základě zvoleného času/periody pro modul SyncArch
SyncMark 1, 10, 0, 0, 1, @archivuj, NONE

//modul pro archivaci proměnné do matice
SyncArch y1, 0, arc_tepl[0,*], arc_time, @archivuj, @archivuj, 3, arc_index, 0x0000
SyncArch y2, 0, arc_tepl[1,*], arc_time, @archivuj, @archivuj, 3, arc_index, 0x0000
SyncArch y3, 0, arc_tepl[2,*], arc_time, @archivuj, @archivuj, 3, arc_index, 0x0000
SyncArch y4, 0, arc_tepl[3,*], arc_time, @archivuj, @archivuj, 3, arc_index, 0x0001

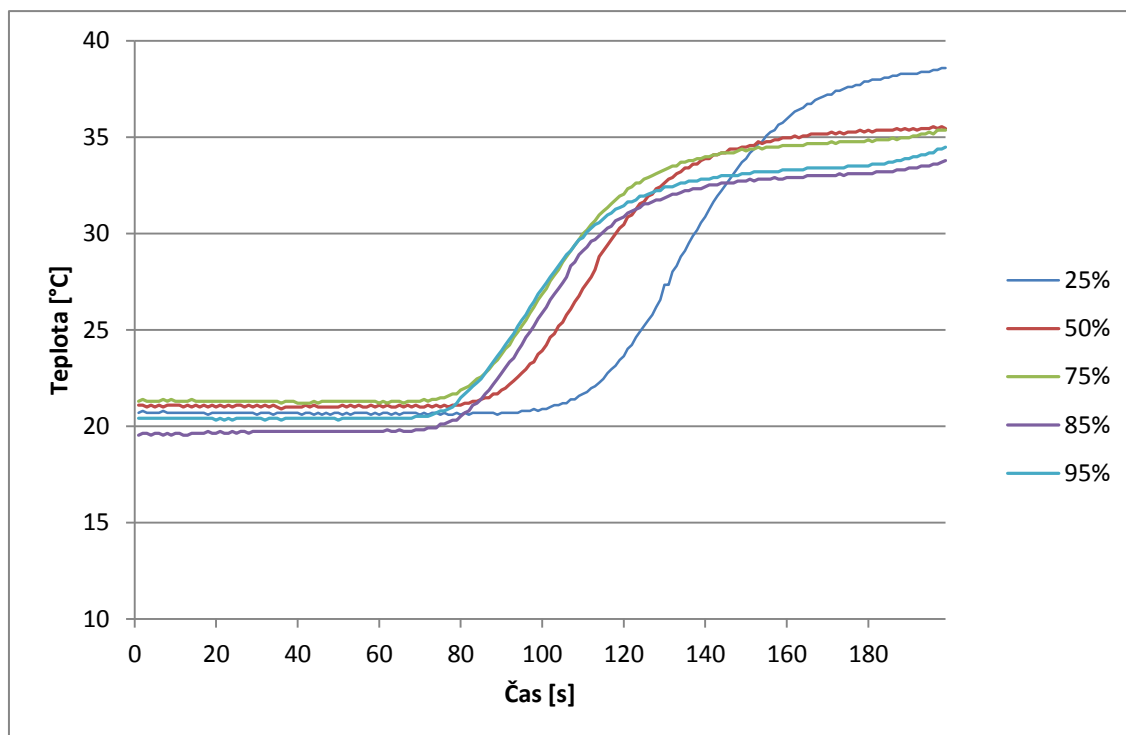
```

Obrázek 6.1.2: Programová část řešící načtení teplot a jejich archivaci

Pro snadnější export dat, jsem vytvořil jednoduchou vizualizaci, která obsahovala prvek PC Archiv, ve kterém se zobrazoval obsah matice arc_tepl. Exportovaná data jsem pak následně upravoval pomocí programu MS Excel.

6.2 Měření přechodové charakteristiky

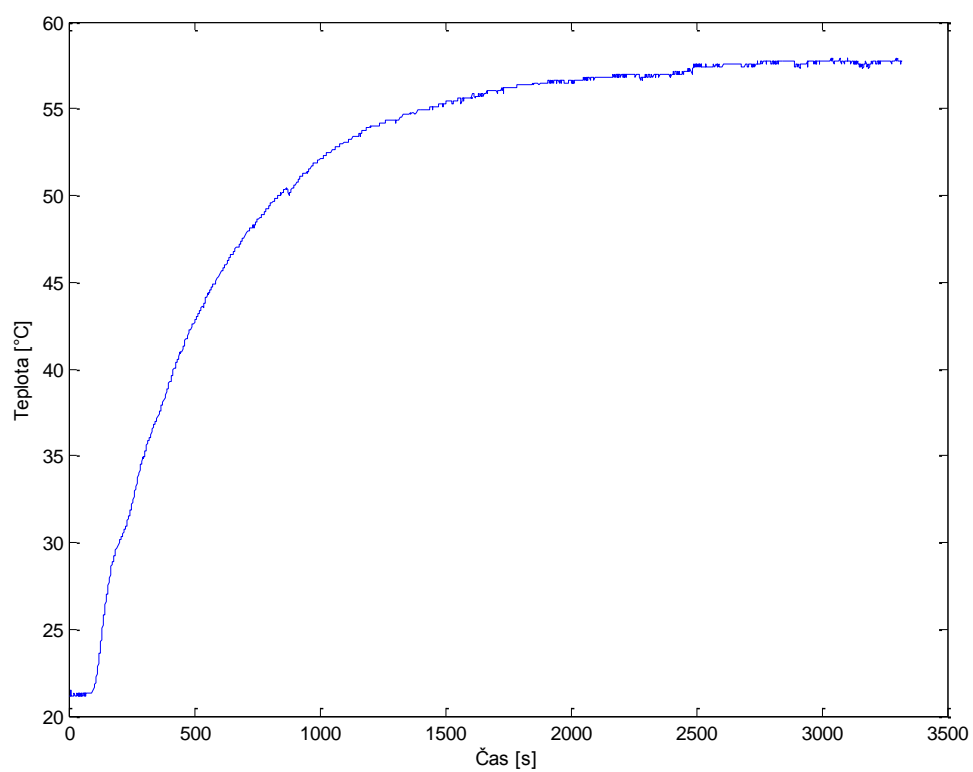
Před měřením samotné přechodové charakteristiky jsem provedl sérii měření pro různé výkonové hladiny čerpadla. Všechna měření probíhala za stejných počátečních podmínek a nastavení. Jedinou proměnnou veličinou byl výkon čerpadla. Porovnání jednotlivých měření je vidět v níže uvedeném grafu (Graf 6.2.1). Měření bylo provedeno na měřené teplotě y2 (teplota před výměníkem).



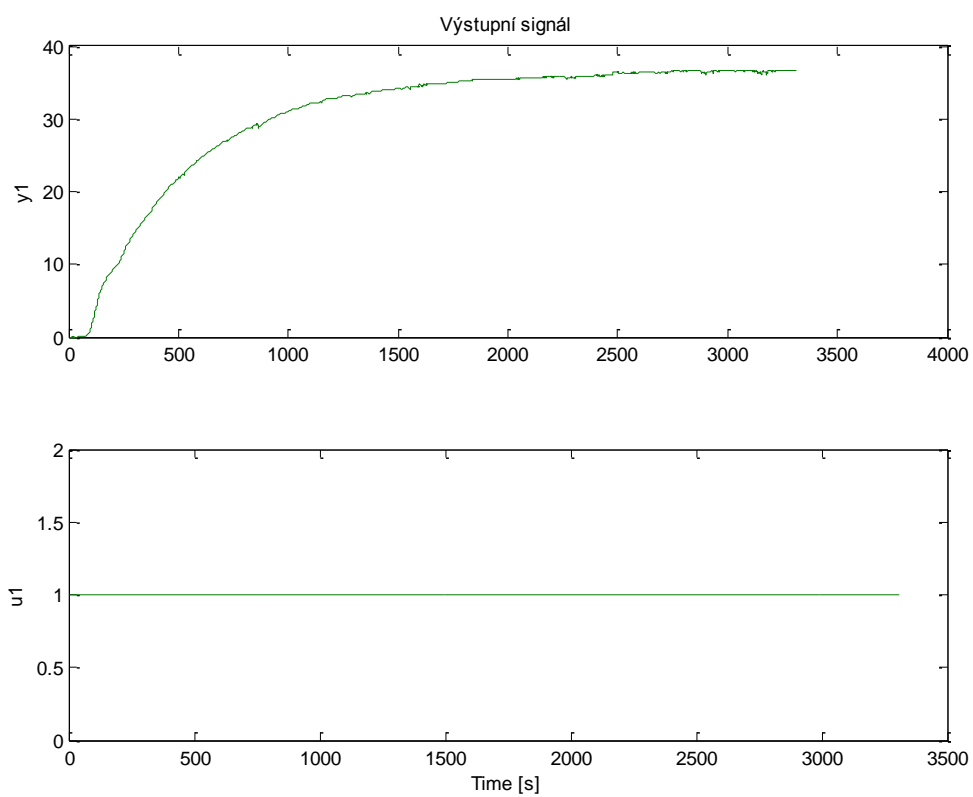
Graf 6.2.1: Porovnání charakteristik při různém výkonu čerpadla

Z grafu lze vyčíst, že s nižším výkonem stoupá dopravní zpoždění, ale zároveň od výkonové hladiny 75% se dopravní zpoždění téměř nemění. Na základě tohoto měření jsem vybral výkonovou hladinu 75%.

Po zisku a nastavení počátečních podmínek jsem změřil přechodovou charakteristiku (Obrázek 6.2.1) průběhu teploty y4 (teplota vystupující z výměníku). Na obrázku 6.2.2 je zobrazená přechodová charakteristika, která je již posunutá do pracovního bodu a tím připravena pro další zpracování.



Obrázek 6.2.1: Přechodová charakteristika



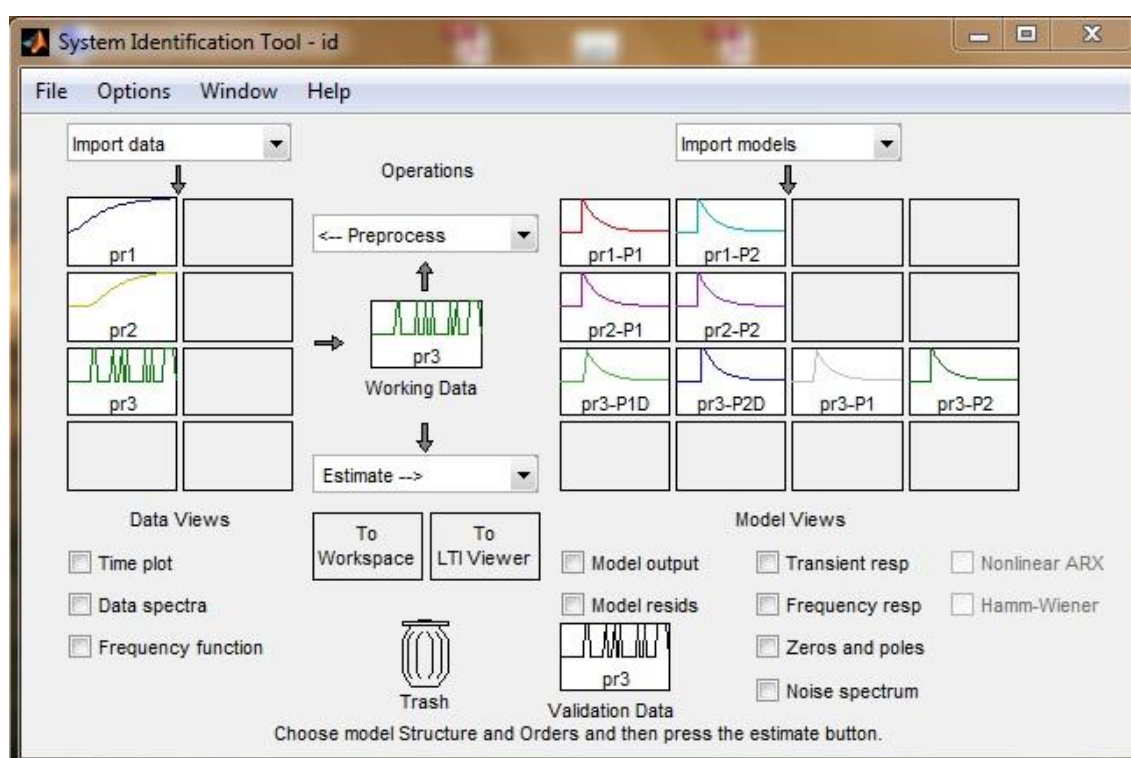
Obrázek 6.2.2: Přechodová charakteristika

6.3 Zpracování přechodové charakteristiky v programu Matlab

Pro zpracování charakteristiky programem Matlab, ji bylo potřeba upravit do podoby jak je na Obrázku 6.2.1. Úprava proběhla v programu MS Excel. První úpravou bylo nahrazení desetinné čárky tečkou. Druhá úprava spočívala v posunutí průběhu do nulových počátečních podmínek.

K identifikaci systému se v Matlabu využívá funkce „ident“. Ta na základě vstupních dat, naměřené odezvy a nastavení několika parametrů nám vrátí přenos aproximovaného systému s procentuální přesností, jak odpovídá aproximovaný systém tomu originálnímu.

Po spuštění funkce ident se otevře základní okno (Obrázek 6.3.1) kde na levé straně jsou vstupní data a na pravé straně aproximované systémy.



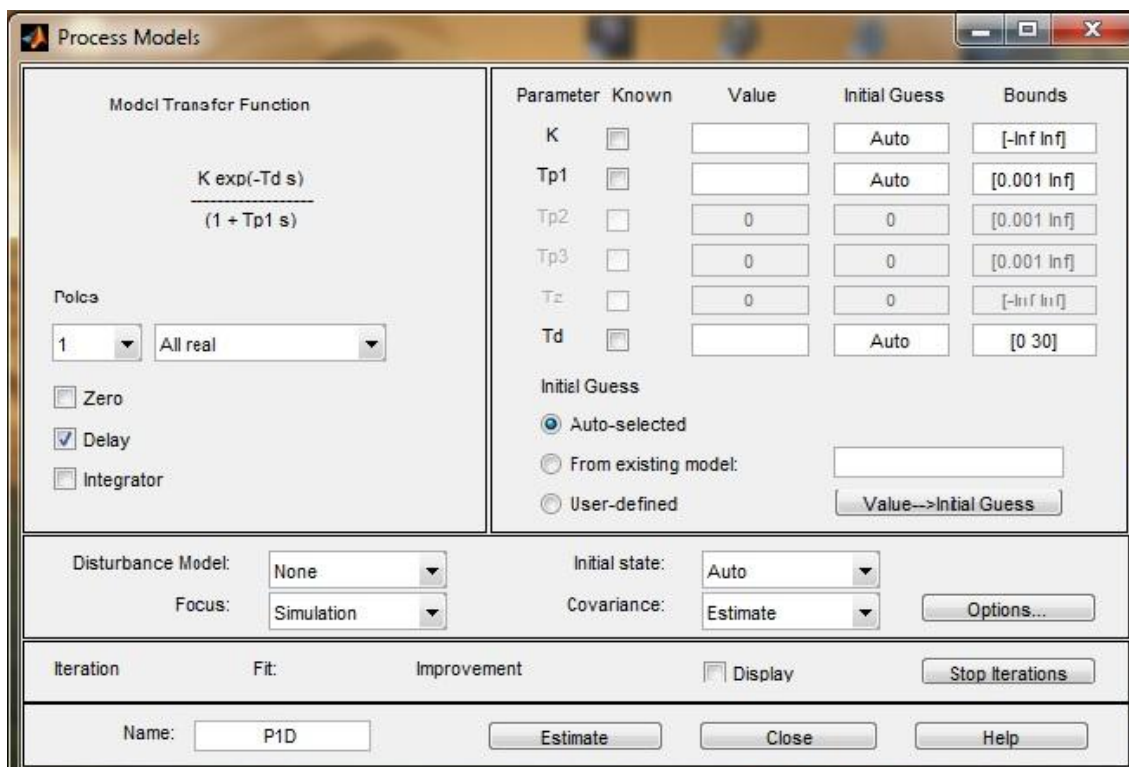
Obrázek 6.3.1: Základní okno funkce ident

Přidání naměřené přechodové charakteristiky se dělá pomocí možnosti „import data“. V nově otevřené nabídce je potřeba nastavit vstupní data, v tomto případě jednotkový skok, výstupní data, naměřená přechodová charakteristika, jméno charakteristiky a počáteční čas, obvykle 0.

V dalším kroku je potřeba přetáhnout přidanou charakteristiku do položky Working data a Validation data. Tím se vybírá, jaká vstupní data budou zpracovávána.

Jako poslední je potřeba vybrat položku „process models“ (Obrázek 6.3.2), která provede výpočet. Nejprve je však nutné nastavit jaký systém a řád systému. Na výběr je systém s dopravním zpožděním, nebo integračním charakterem. Řád systému lze zvolit mezi prvním až třetím řádem. Poté už stačí jenom nastavit jméno aproximovaného systému a spustit výpočet. Je zde i možnost omezit interval hodnot, se kterým se bude počítat, nebo i provést vlastní „nástřel“ hodnot. Po dokončení výpočtu

se spočítaný průběh zobrazí v pravé části okna funkce ident. Pokud vybereme možnost „model output“, tak se zobrazí graf, kde bude zobrazený aproximovaný průběh a naměřený průběh. V této položce bude zobrazena i procentuální hodnota, jak přesný aproximovaný systém je.



Obrázek 6.3.2: Okno položky process models

Tímto způsobem jsem aproximoval čtyři systémy (Obrázky 6.3.3 – 6.3.6):

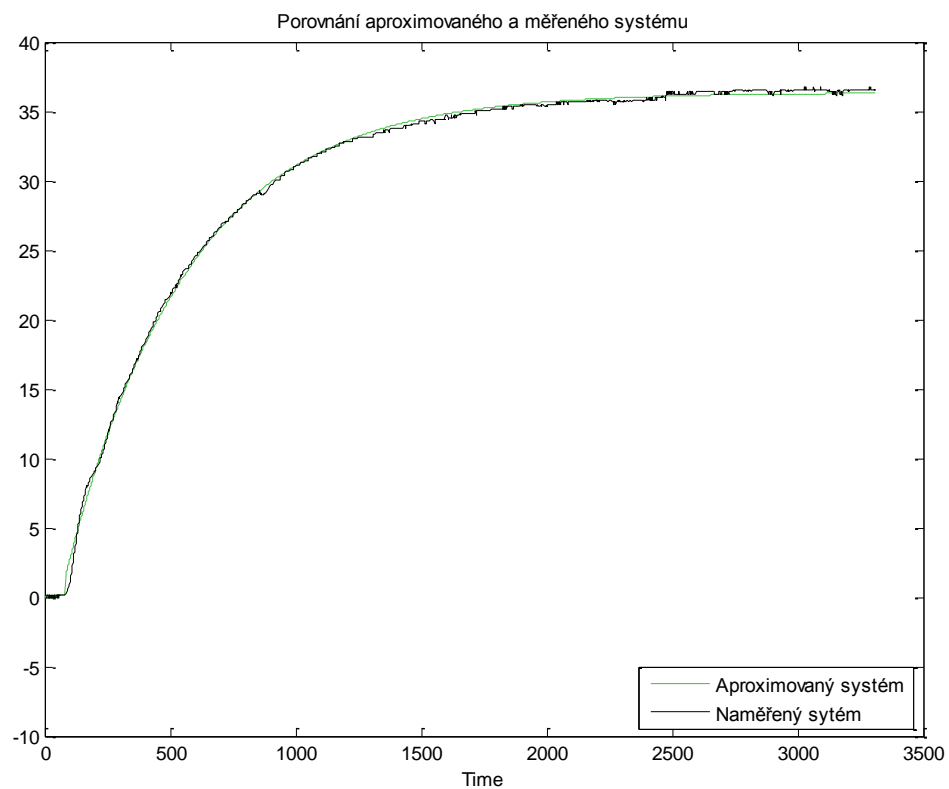
$$\text{První řád s dopravním zpožděním: } G(s) = \frac{36,345}{(1+486,39 \cdot s)} * e^{(-83 \cdot s)} \quad (6.1)$$

$$\text{Druhý řád s dopravním zpožděním: } G(s) = \frac{36,41}{(1+500,55 \cdot s) \cdot (1+21,657 \cdot s)} * e^{(-100 \cdot s)} \quad (6.2)$$

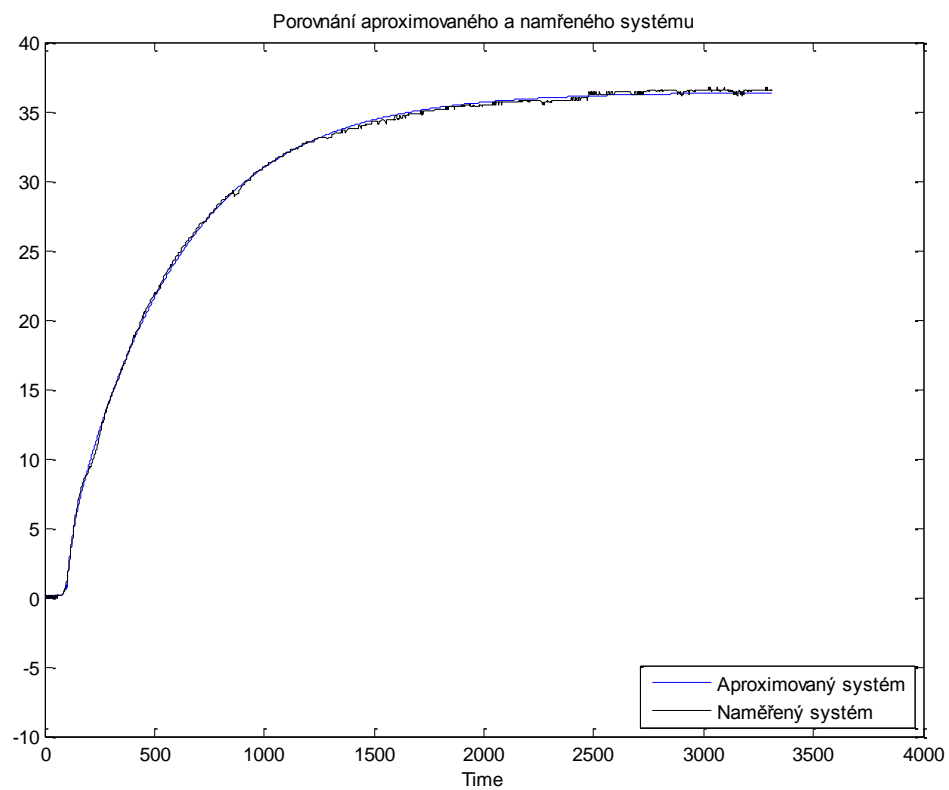
$$\text{První řád bez dopravního zpoždění: } G(s) = \frac{36,436}{1+503,62 \cdot s} \quad (6.3)$$

$$\text{Druhý řád bez dopravního zpoždění: } G(s) = \frac{6389}{(1+2,1064 \cdot 10^7 \cdot s) \cdot (1+486 \cdot s)} \quad (6.4)$$

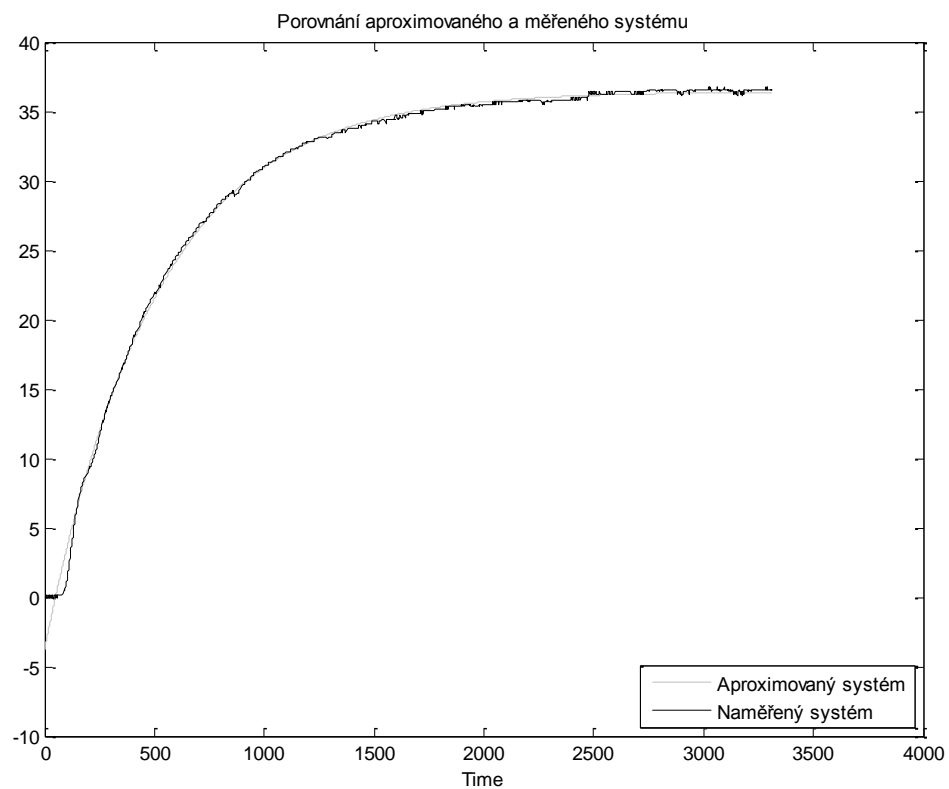
Procentuální přesnost přenosů s dopravním zpožděním byla vyšší než 97%. Druhé dva průběhy, bez dopravního zpoždění, dosáhly přesnosti trochu nižší, přibližně 95%.



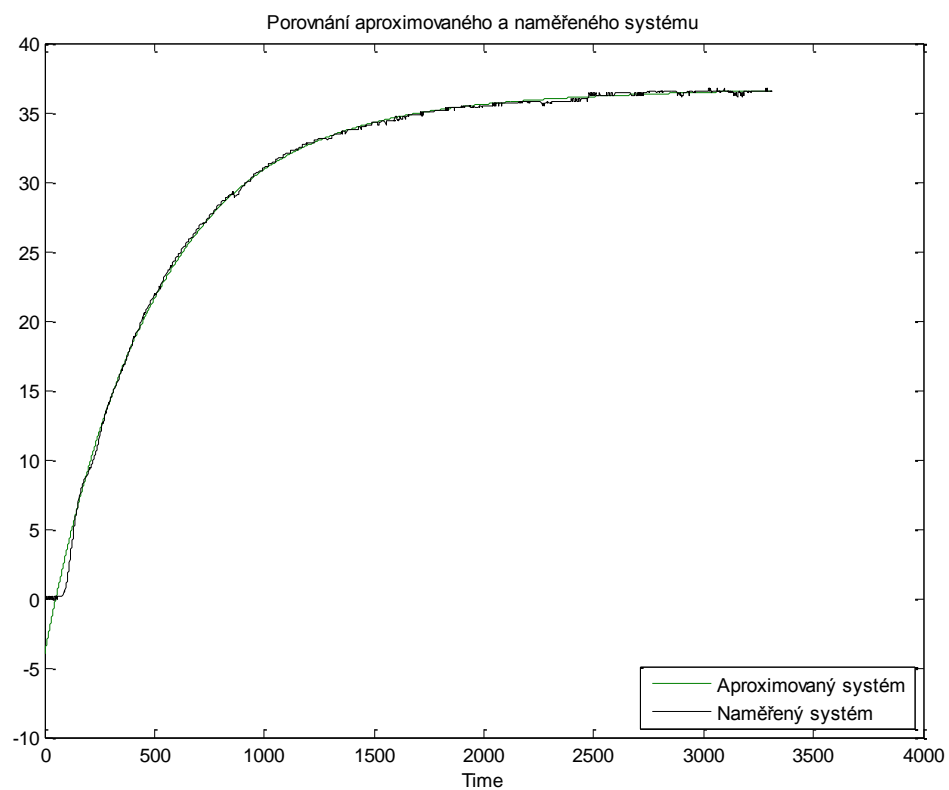
Obrázek 6.3.3: Aproximovaný systém prvního řádu s dopravním zpožděním, přesnost 97,03%



Obrázek 6.3.4: Aproximovaný systém druhého řádu s dopravním zpožděním, přesnost 97,79%



Obrázek 6.3.5: Aproximovaný systém prvního řádu bez dopravního zpoždění, přesnost 95,36%



Obrázek 6.3.6: Aproximovaný systém druhého řádu bez dopravního zpoždění, přesnost 95,47%

7. Návrh regulátoru

Při návrhu parametrů pro regulátor jsem vycházel z přenosu soustavy prvního řádu s dopravním zpožděním (6.3.1) a z přenosu soustavy druhého řádu s dopravním zpožděním (6.3.2). Pro řešení této úlohy jsem zvolil dvě možnosti regulace. Řízení pomocí PID regulátoru a Smithova prediktoru, který se využívá v soustavách s dopravním zpožděním.

7.1 Návrh PID regulátoru

Protože téměř všechny návrhové metody vycházejí z hodnot doby náběhu (T_n) a doby průtahy (T_u), tak jsem před samotným návrhem tyto doby vypočítal. K tomu jsem využil skript (Příloha III) vytvořený v programu Matlab, který na základě vloženého přenosu soustavy tyto doby spočítá. Výpočet pracuje pouze s přenosy bez dopravního zpoždění, proto k výsledným dobám je potřeba dopravní zpoždění přičíst, abychom dostali správné výsledné doby.

Tabulka 7.1.1: Doby náběhu a průtahy pro přenos prvního a druhého řádu

	1. řád	2. řád
T_u	83	116,3802
T_n	568,4049	675,4358

Pro srovnání jsem použil několik metod. První metodu, kterou jsem použil, je metoda Zieglera a Nicholse. Tento postup je nejstarší a byl vytvořen především pro regulaci chemických procesů. Tato pravidla byla optimalizována pro dobré potlačení poruch, a proto jsou nevhodná a prakticky nepoužitelná pro sledování skokových změn žádané veličiny. Ziegler a Nichols zavedli dvě možnosti výpočtu. Metoda ustálených kmitů, kde se systém přivádí na mez stability a zjišťují se kritické hodnoty, druhá metoda je založena na známé přechodové odezvě. V tomto případě jsem využil metodu na známé přechodové odezvě.

Tabulka 7.1.2: Výpočty parametrů regulátoru

Regulátor	r_0	T_i	T_d
P	$1/(K*(T_u/T_n))$		
PI	$0,9/(K*(T_u/T_n))$	$3*T_u$	
PID	$1,2/(K*(T_u/T_n))$	$2*T_u$	$0,5*T_u$

Tabulka 7.1.3: Parametry regulátoru

ZN	1. řád	2. řád
r_0	0,226111	0,191278
T_i	166	232,7604
T_d	41,5	58,1901

Další použitou metodou je Kuhnova metoda, pravidlo souhrnné časové konstanty. Zde se využívá tzv. časová souhrnná konstanta, která je dána součtem všech časových konstant a dopravního zpoždění soustavy. Tato pravidla jsou relativně konzervativní. Regulační pochod je obvykle málo kmitavý, s dobou regulace přibližně

stejnou v odezvě na změnu žádané hodnoty i na vstupní poruchu. Někdy však může být až zbytečně pomalý. Ve srovnání s jinými nastavovacími postupy dávají dobré výsledky zejména u PI regulátorů, méně již u PID regulátorů.

Pro přenos ve tvaru:

$$G(s) = K_s \frac{(T_{1N}s + 1)(T_{2N}s + 1) \cdots (T_{mN}s + 1)}{(T_1s + 1)(T_2s + 1) \cdots (T_ns + 1)} e^{-T_D s} \quad (7.1.1)$$

Lze tzv. souhrnnou časovou konstantu definovat jako:

$$T_\Sigma = T_1 + T_2 + \cdots + T_n - T_{1N} - T_{2N} - \cdots - T_{mN} + T_D \quad (7.1.2)$$

Na základě hodnot zesílení (K) a souhrnné časové konstanty lze vypočítat parametry regulátoru takto:

Tabulka 7.1.4: Výpočty parametrů regulátoru

Regulátor	r0	Ti	Td
P	1/K		
PD	1/K		0,33*T _s
PI	0,5/K	0,5*T _s	
	1/K	0,7*T _s	
PID	1/K	0,66*T _s	0,167*T _s
	2/K	0,8*T _s	0,194*T _s

Tabulka 7.1.5: Parametry regulátoru

KUHN	1. řád	2. řád
r0	0,055028	0,05493
Ti	455,512	497,7656
Td	110,4617	120,7082

Nejvhodnější metodou pro nastavení parametrů regulátoru pro tuto soustavu jsou pravidla Chien, Hrones a Reswicka. Tato pravidla byla odvozena za předpokladu, že regulovaný systém je popsán přenosem prvního řádu s dopravním zpožděním:

$$G(s) = \frac{K \cdot e^{(T_D \cdot s)}}{\tau \cdot s + 1} \quad (7.1.3)$$

Na rozdíl od předchozích však umožňují výběr ze dvou variant regulačního pochodu: aperiodický a s překmitem 20% a také volbu, zda cílem regulace je sledování změn žádané hodnoty nebo potlačení poruch na vstupu soustavy.

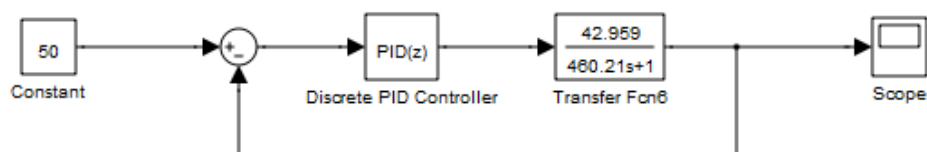
Regulátor	Aperiodický regulační pochod		Překmit 20%	
	Žádaná hodnota	Porucha	Žádaná hodnota	Porucha
P	$r_o=0,3 \tau/(KT_D)$	$r_o=0,3 \tau/(KT_D)$	$r_o=0,7 \tau/(KT_D)$	$r_o=0,7 \tau/(KT_D)$
PI	$r_o=0,35 \tau/(KT_D)$ $T_i=1,2 \tau$	$r_o=0,6 \tau/(KT_D)$ $T_i=4T_D$	$r_o=0,6 \tau/(KT_D)$ $T_i=\tau$	$r_o=0,7 \tau/(KT_D)$ $T_i=2,3T_D$
PID	$r_o=0,6 \tau/(KT_D)$ $T_i=\tau$ $T_d=0,5T_D$	$r_o=0,95 \tau/(KT_D)$ $T_i=2,4T_D$ $T_d=0,42T_D$	$r_o=0,95 \tau/(KT_D)$ $T_i=1,35 \tau$ $T_d=0,47T_D$	$r_o=1,2 \tau/(KT_D)$ $T_i=2T_D$ $T_d=0,42T_D$

Obrázek 7.1.1: Výpočty parametrů regulátoru

Tabulka 7.1.6: Parametry regulátoru

CH. H. R.	1. řád	2. řád
r0	0,113054	0,095639
Ti	568,4049	675,4358
Td	41,5	58,1901

Nejjednodušší nastavení lze provést pomocí programu Matlab. V tomto případě se do simulace (Obrázek 7.1.2) vloží blok $PID(z)$, který sám zvolí nejvhodnější parametry. V jeho nastavení (Obrázek 7.1.3) si lze zvolit typ regulátoru, formu regulátoru (paralelní, sériový), nastavit zda bude spojitý nebo diskrétní atd. Po kliknutí na položku Tune, se provede optimalizovaný výpočet regulovaného průběhu a nastavení parametrů. Průběh regulované veličiny lze ručně doladit. Pomocí jednoduchého rozhraní (Obrázek 7.1.4), můžeme měnit dobu regulačního pochodu, a tím si zvolit nejvhodnější regulační průběh, který potřebujeme pro danou úlohu.



Obrázek 7.1.2: Simulační schéma s blokem PID

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PID** Form: **Parallel**

Time domain:

☐ Continuous-time

☒ Discrete-time

Discrete-time settings

Integrator method: **Forward Euler**

Filter method: **Forward Euler**

Sample time (-1 for inherited): **1**

Main **PID Advanced** Data Types State Attributes

Controller parameters

Proportional (P): **0.328779220234729** [Compensator formula](#)

Integral (I): **0.000952964270174082**

Derivative (D): **-5.3177788917039**

Filter coefficient (N): **0.0598881639283209**

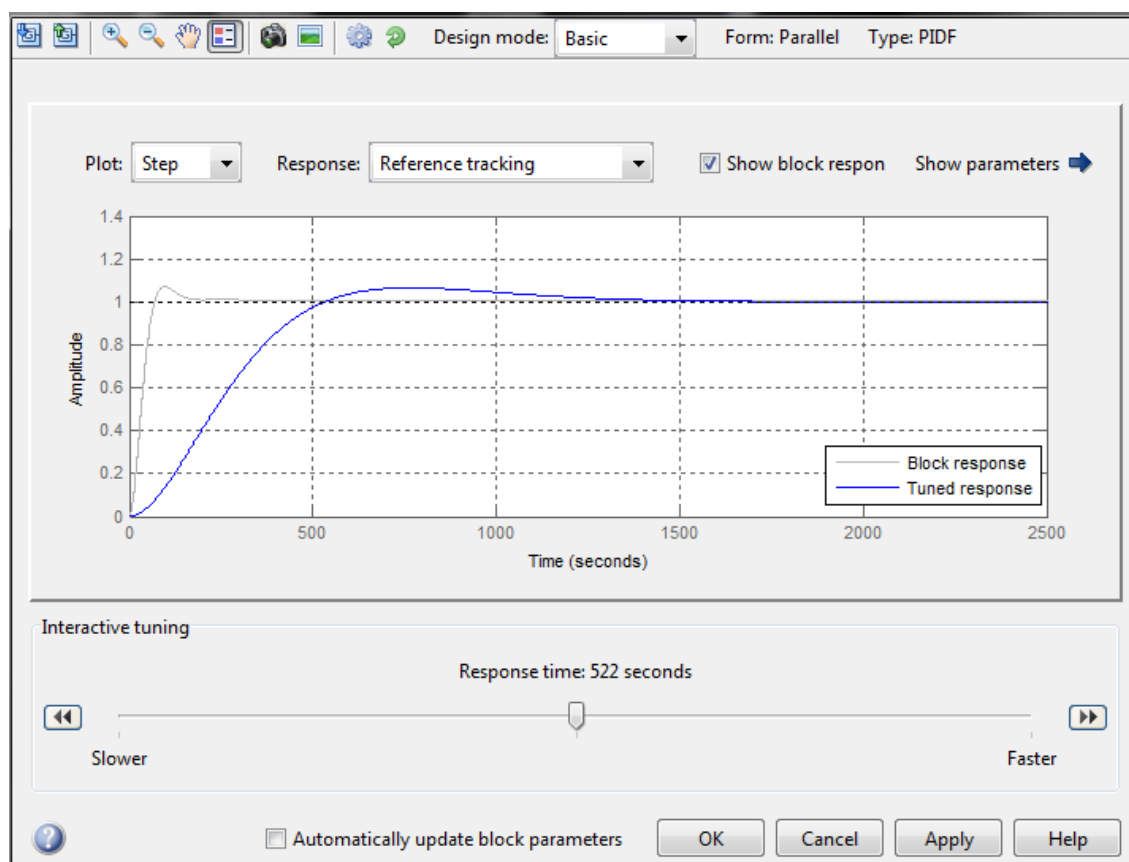
$$P + I \cdot T_s \frac{1}{z-1} + D \frac{N}{1 + N \cdot T_s \frac{1}{z-1}}$$

Tune...

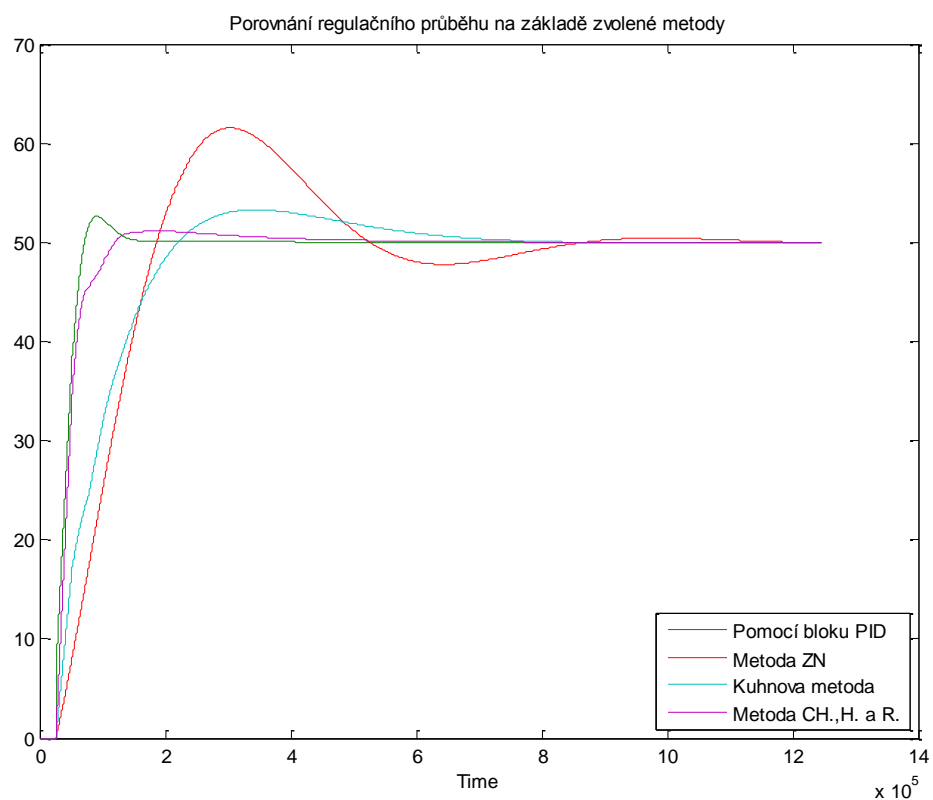
Initial conditions

OK **Cancel** **Help** **Apply**

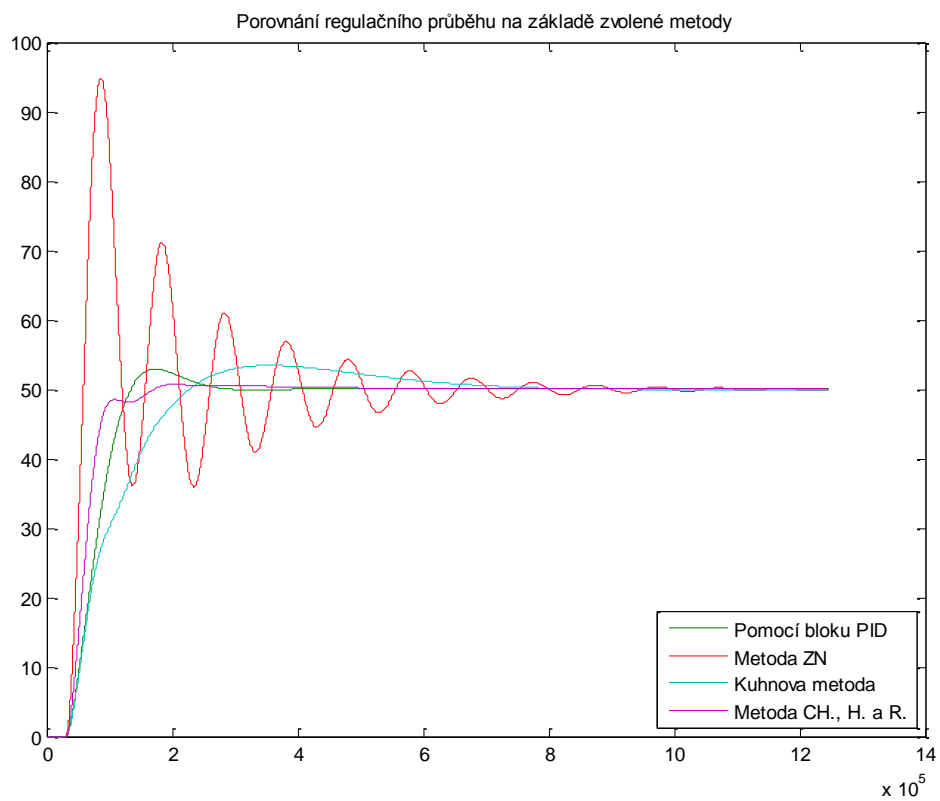
Obrázek 7.1.3: Záložka nastavení Bloku PID



Obrázek 7.1.4: Rozhraní s možností úpravy regulačního průběhu



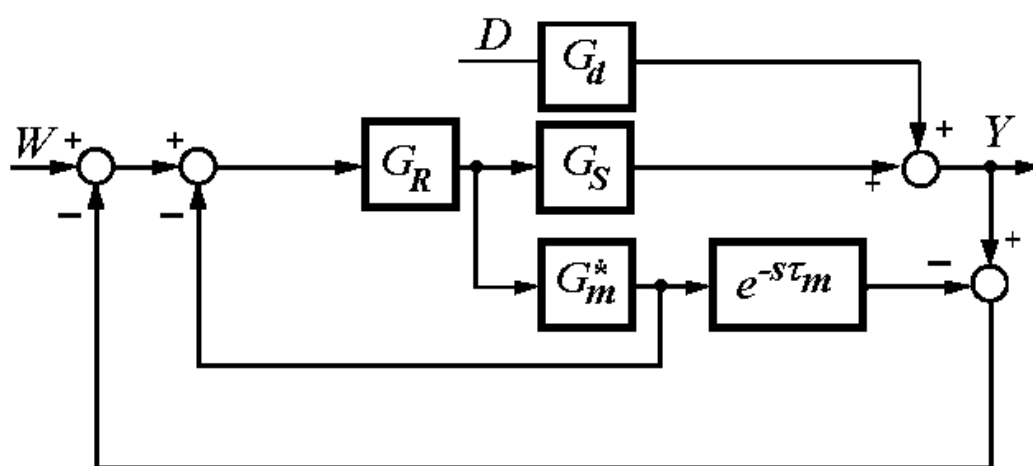
Obrázek 7.1.5: Porovnání regulačních průběhů systému prvního řádu



Obrázek 7.1.6: Porovnání regulačních průběhů systému druhého řádu

7.2 Návrh Smithova prediktoru

Smithův prediktor se používá k řízení stabilních systémů s dopravním zpožděním. Schéma je vidět na obrázku 7.2.1. Na rozdíl od jednoduchého regulačního obvodu, který obsahuje jako prvky pouze řízený systém G_S a regulátor G_R , obsahuje struktura Smithova prediktoru i další prvky. A to model řízeného procesu bez dopravního zpoždění G_m^* a odhad skutečného dopravního zpoždění řízeného procesu (tedy dopravního zpoždění modelu), označený jako $e^{-s\tau_m}$.



Obrázek 7.2.1: Schéma Smithova prediktoru

Parametry pro Smithův prediktor jsem určil pomocí experimentální metody (ruční nastavení) v simulaci v programu Matlab.

Tabulka 7.2.1: Parametry Smithova prediktoru

Smith. P.	1. řád
r0	0,8
Ti	1400
Td	30

8. Programové řešení regulátoru

8.1 PID regulátor

Velikou výhodou vývojového prostředí DetStudio je, že jako mnoho jiných funkcí, má integrovaný i PID regulátor. Práce s tímto blokem je jednoduchá a intuitivní, stačí založit pouze pět proměnných, které se u bloku PID nastaví jako vstupy a výstup.

PID `tepl_zadana`, `teplota_mere`, `pid_out`, `pid_rezim`, `pid_para`
kde:

`tepl_zadana` – proměnná s hodnotou žádané hodnoty veličiny

`teplota_mere` – proměnná s hodnotou měřené veličiny

`pid_out` – proměnná s hodnotou akčního zásahu

`pid_rezim` – proměnná, která obsahuje režim činnosti a volby regulátoru
(podrobně vysvětleno v nápovědě programu DetStudio)

`pid_para` – matice s parametry regulátoru – rozsah, konstanty...
(podrobně vysvětleno v nápovědě programu DetStudio)

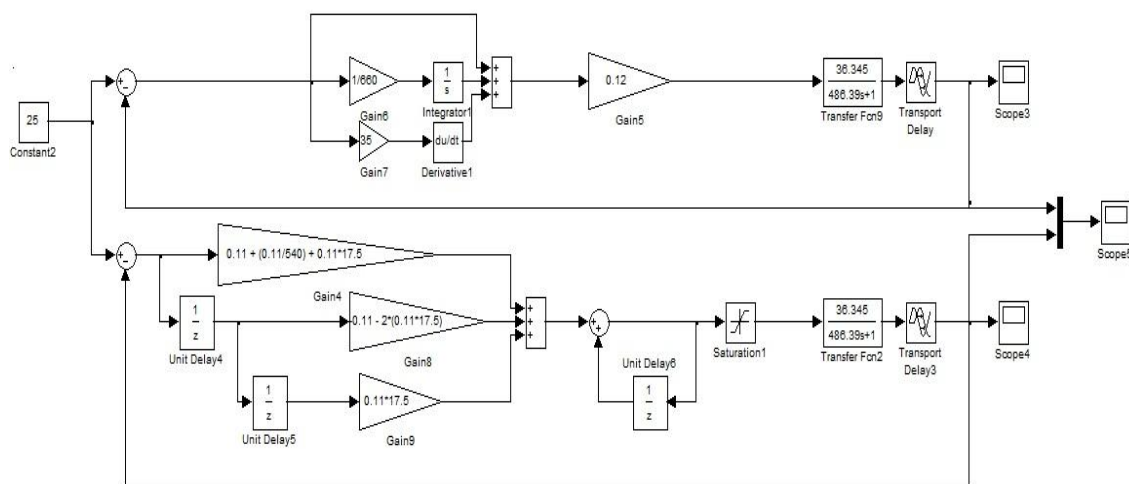
Bohužel jsem během mé práce s tímto blokem nemohl pracovat, protože výstup regulátoru byl pouze 0 nebo 100%. Proč tomu tak bylo, se mi nepodařilo zjistit, i přes to, že jsem několikrát zkontroloval nastavení, které bylo v pořádku. Zkoušel jsem regulátor aplikovat na jednoduchou aplikaci, kde jsem nastavil žádanou hodnotu a měnil měřenou hodnotu, ale výsledek byl pořád stejný.

Z tohoto důvodu jsem zvolil možnost naprogramovat algoritmus pro výpočet akčního zásahu regulátoru. Zde jsem narazil na problém, který spočíval v tom, že je mnoho zdrojů, které uvádějí mnoho různých výpočtů. Během řešení této překážky jsem vyzkoušel více jak deset možností. V průběhu hledání a zkoušení jsem došel k závěru, že čím jednodušší výpočty, tím výsledná regulace byla přesnější. Proto jsem zvolil následující výpočet:

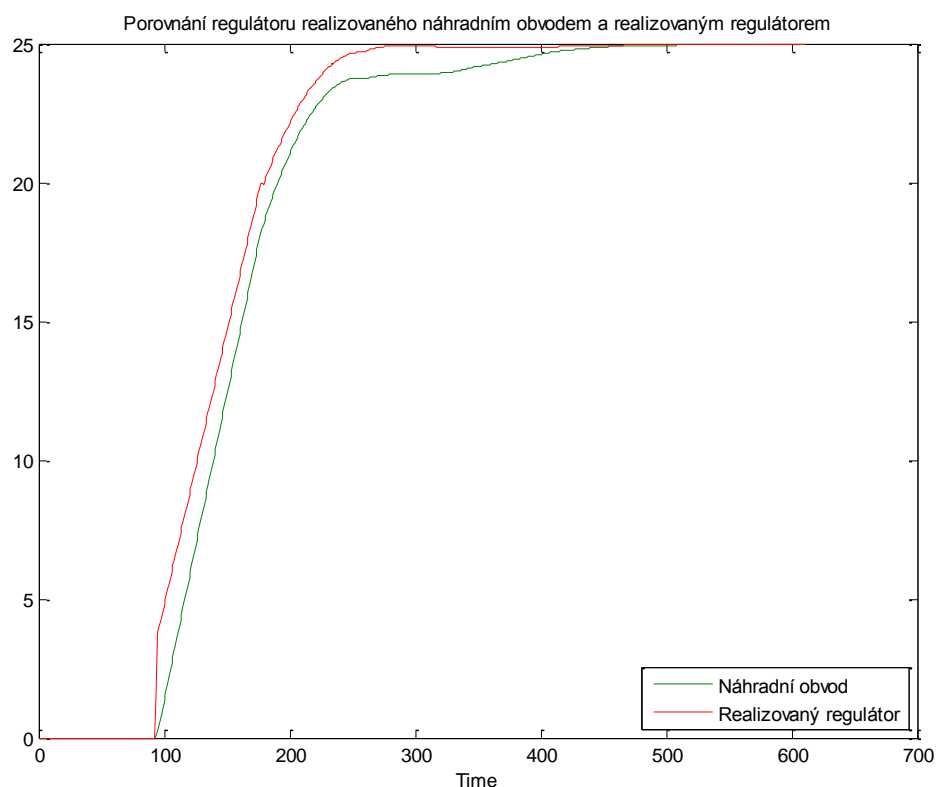
$$\Delta u = (K + (K/T_i) + K \cdot T_d) * e_{(k)} - (K + 2 * (K \cdot T_d)) * e_{(k-1)} + (K \cdot T_d) * e_{(k-2)} \quad (8.1.1)$$

$$u_{(k)} = 100 * (u_{(k-1)} + \Delta u) \quad (8.1.2)$$

Parametry pro PID regulátor jsem zvolil na základě metody Chien, Hrones a Reswicka pro systém prvního řádu (Tabulka 7.1.6). Hodnoty, které jsem touto metodou vypočítal, jsem následně pomocí simulace doladil, aby výsledný regulační průběh byl bez překmitu. Nové parametry jsou vidět na Obrázku 8.1.1, na kterém je simulační schéma, kde je porovnání regulace pomocí náhradního obvodu PID regulátoru a regulátoru realizovaného výše uvedeným výpočtem. Při přechodu na číslicový výpočet, bylo opět potřeba parametry upravit, proto se parametry na Obrázku 8.1.1 liší. Výsledné průběhy jsou vidět na Obrázku 8.1.2, kde je zobrazeno jejich porovnání, na kterém lze vidět, že výsledné regulační průběhy se liší jen nepatrně, a výše uvedený výpočet lze použít.

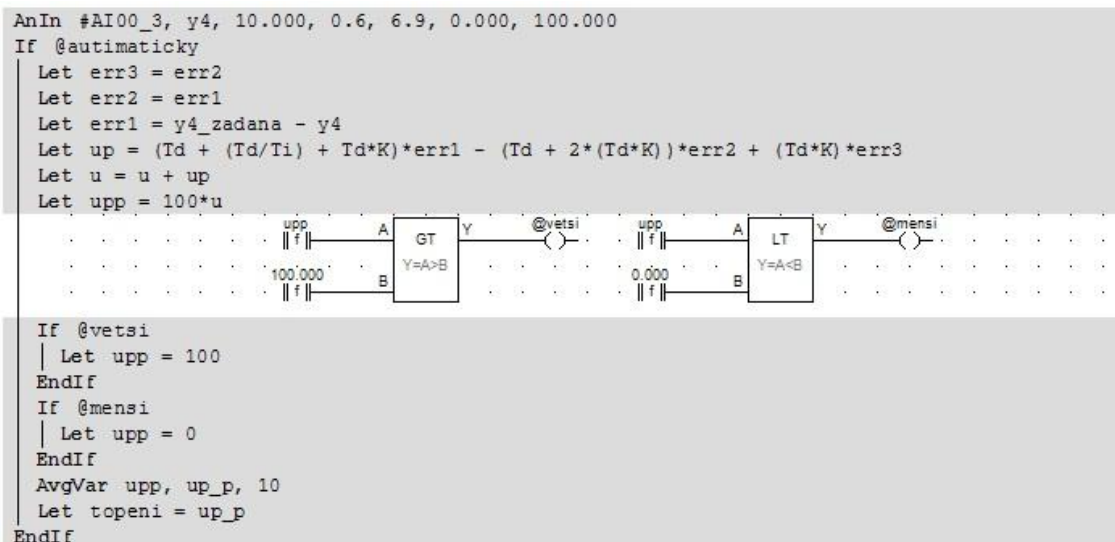


Obrázek 8.1.1: Simulační schéma pro porovnání výpočtů akčního zásahu



Obrázek 8.1.2: Porovnání regulačních průběhů

Programové řešení regulátoru je zobrazeno na Obrázku 8.1.3. Program regulátoru se skládá ze tří hlavních částí. První je výpočet regulačních odchylek a výpočet akčního zásahu. Druhá část porovnává výsledný akční zásah, jestli je v rozsahu 0 – 100, pokud není, nastaví se mez 0 nebo 100. Aby se předešlo kolísání hodnot akčního zásahu, tak se v poslední části programu vypočítává klouzavý průměr z 10-ti posledních hodnot. Teprve zprůměrovaná hodnota se zapisuje do proměnné, která je vstupem PWM modulace, která řídí spínání průtokového ohřívače. Podrobný popis a význam jednotlivých proměnných je zobrazen v Příloze V.

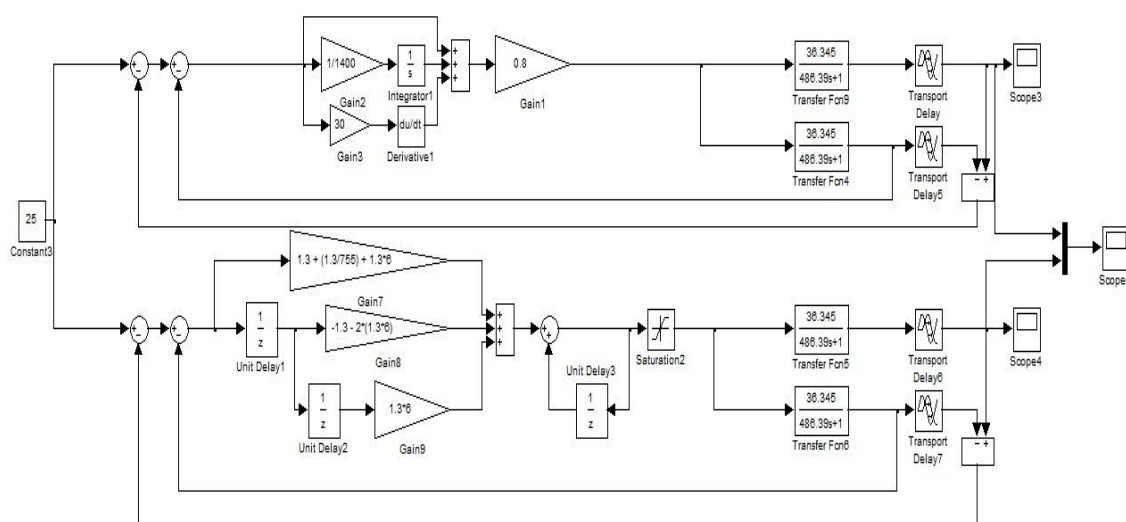


Obrázek 8.1.3: Programové řešení PID regulátoru

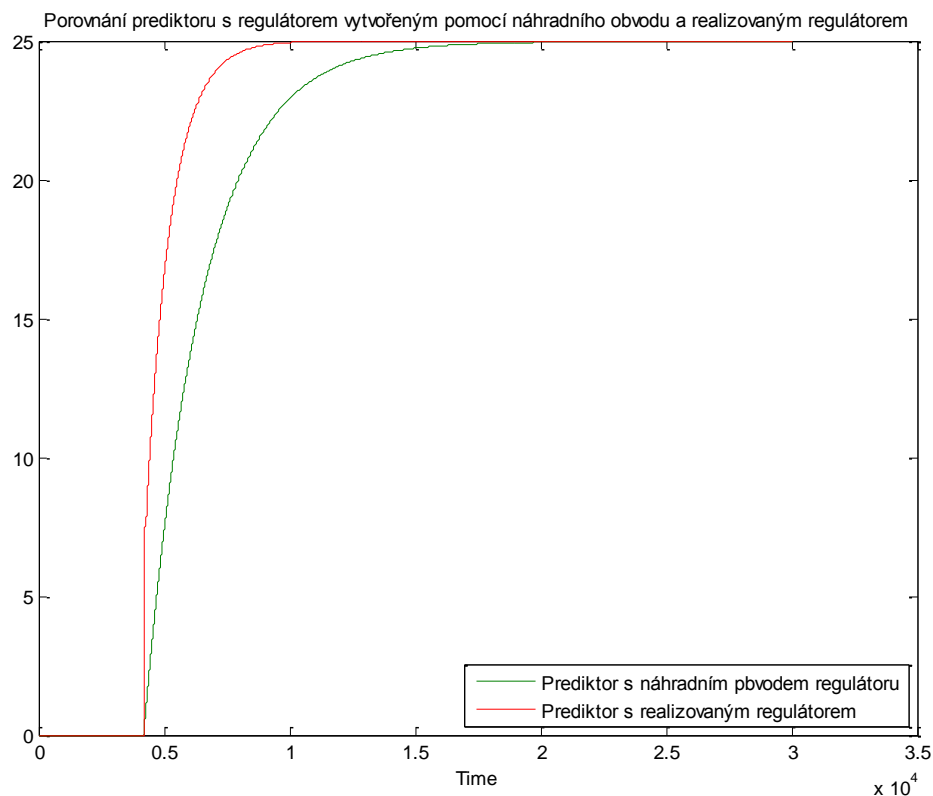
8.2 Smithův prediktor

Smithův prediktor v DetStudiu není implementován, jako je tomu u jiných konkurenčních výrobců, což jsem shledal jako jednu z mála nevýhod, na které by mohl výrobce zapracovat.

Protože prediktor obsahuje PID regulátor, tak jsem opět pomocí simulace porovnal prediktor obsahující náhradní obvod pro PID regulátor a prediktor s navrženým číslicovým výpočtem. Prediktor realizovaný náhradním obvodem měl nastavené parametry, které jsou v tabulce 9. Kvůli číslicovému výpočtu bylo opět potřeba upravit parametry. V simulačním schématu (Obrázek 8.2.1) je vidět realizace prediktoru a zároveň nové parametry pro Smithův prediktor. Na Obrázku 8.2.2 je vidět porovnání průběhů regulovaných veličin.



Obrázek 8.2.1: Simulační schéma Smithova prediktoru s náhradním obvodem PID regulátoru a s číslicovým výpočtem PID regulátoru



Obrázek 8.2. 2: Porovnání regulačních průběhů Smithova prediktoru

Programové řešení Smithova prediktoru (Příloha IV) je oproti PID regulátoru složitější. Skládá se už z více částí. Z PID regulátoru, diferenční rovnice přenosu systému, archivace dat, kterou se simuluje dopravní zpoždění hodnot, a následných výpočtů pomocných proměnných a regulačních odchylek. Uvedený program prediktoru je spíše informativní jak by se dal v DetStudiu zrealizovat. Naprogramovaný prediktor nemá dokonalý výpočet, který se mi bohužel nepovedlo zdokonalit, aby byl Smithův prediktor použitelný pro řešení této úlohy.

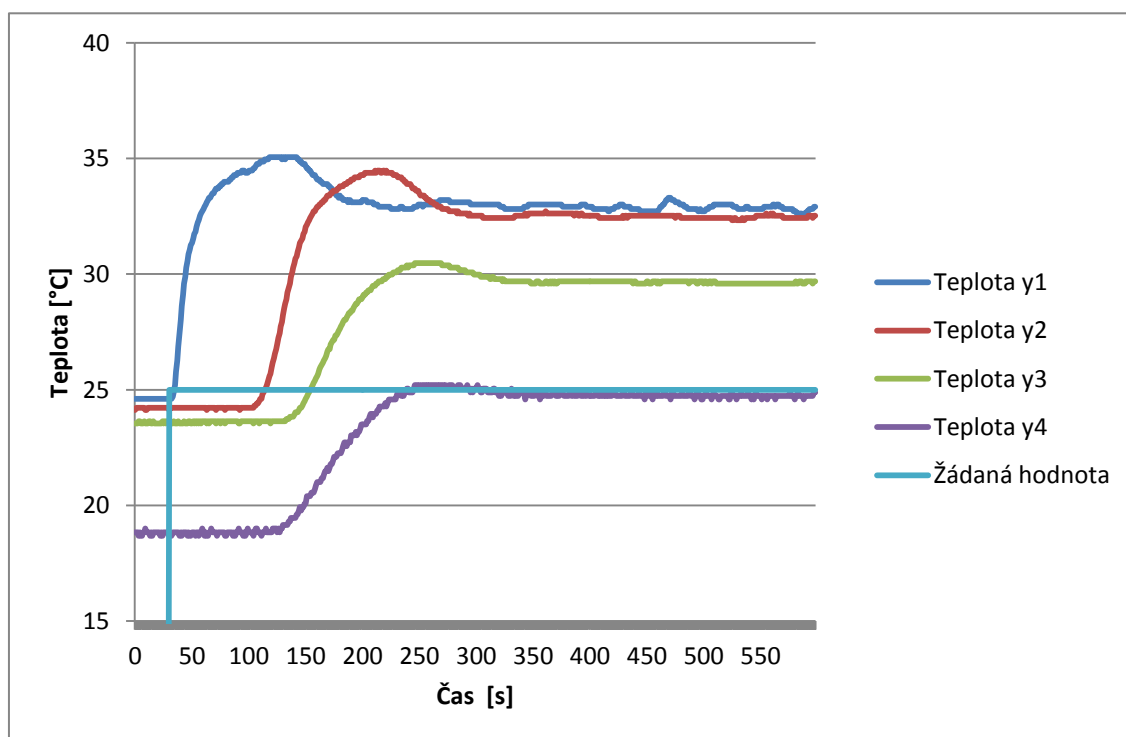
Regulovaný průběh měl veliký překmit, který se mi nepodařilo eliminovat. Udržování žádané hodnoty bylo relativně dobré, bez odchylek větších než přibližně $0,5^\circ\text{C}$, ale kolísavé. Nedokonalost se především projevila při spuštění poruchy, kdy prediktor nebyl schopen doregulovat zpět na žádanou hodnotu. Regulační průběh při spuštěné chybě hodně kolísal a byl nepřesný.

Problém se pravděpodobně skrývá v části, kde se vypočítává teoretická hodnota výstupu pomocí diferenční rovnice a následně se zpožďuje o hodnotu dopravního zpoždění. To by mohlo být způsobeno chybným výpočtem pomocných proměnných, a tím způsobený chybný výpočet regulačních odchylek, což by vedlo ke špatnému akčnímu zásahu regulátoru.

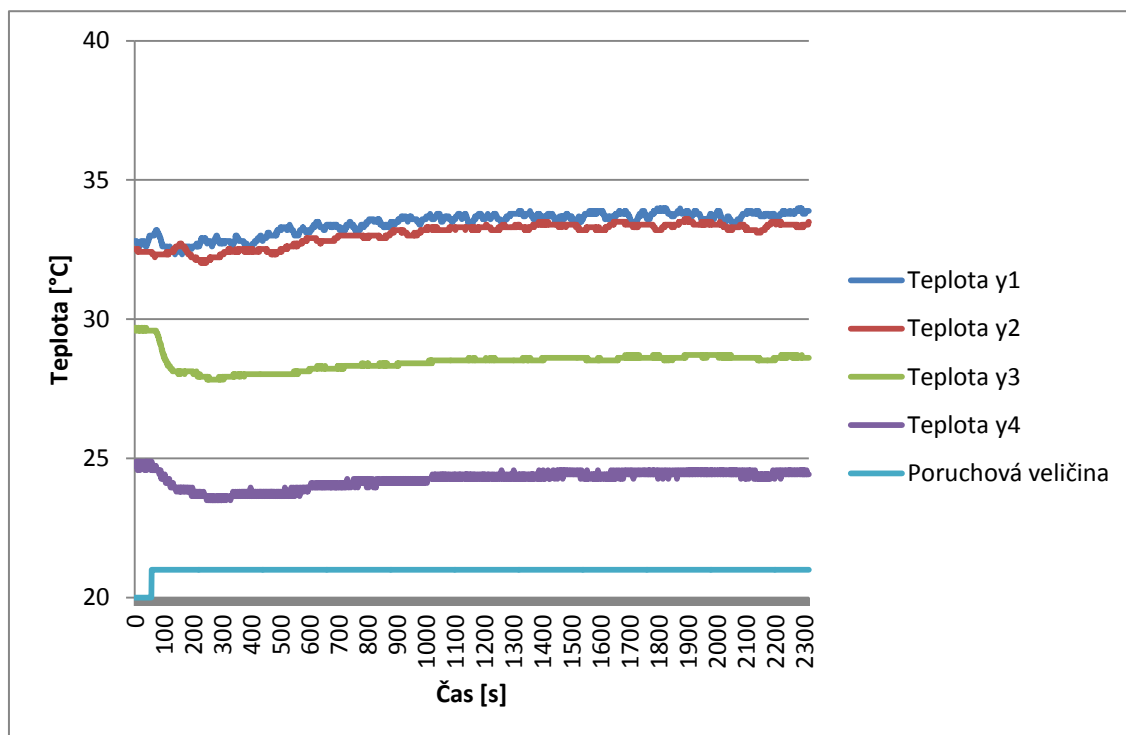
9. Vyhodnocení regulace PID regulátoru

Na níže uvedeném grafu (Graf 9.1) je zobrazen průběh regulace teploty y4. V grafu jsou zobrazeny i zbylé teploty. Žádaná hodnota byla nastavena na 25 °C. V grafu lze vidět, že ustálené hodnoty se dosáhne přibližně za 6 minut od spuštění měření. Co se týče kvality regulace, tak je vidět, že se dosáhne drobného překmitu, a ustálená hodnota se pohybuje několik desetín stupně pod žádanou hodnotou. Toto se dá považovat za jistou míru tolerance, protože se nepřekračuje hranice vyšší jak 0,5 °C.

V průběhu regulace byla puštěna chyba (sepnutí ventilátoru č. 2 na 100%), což lze vidět na průběhu teploty y3 přibližně v čase 11 minut. Zde je vidět, že regulátor má pomalejší reakci na tuto chybu, což je trochu nežádoucí a odezva by měla být rychlejší. To je způsobeno nastavením parametrů a zvolenou metodou, která není specializována na sledování poruchové veličiny.



Graf 9.1: Průběh regulace teploty y4 modelu soustavy



Graf 9.2: Odezva na spuštěnou chybu

10. Časové plány

Časové plány je možné v DetStudiosu realizovat v programové části pomocí modulů DayPlan / DayPlan2. Jejich parametrizace je pak možná také v prostředí ViewDet. Moduly DayPlan / DayPlan2 umožňují plánování hodnoty pro jednotlivé dny v týdnu. Pomocí těchto modulů lze vytvořit zvláštní průběhy plánu pro každý den v týdnu, resp. pro skupinu dnů. Tyto moduly pro svoji činnost vyžadují zadat matici časových zlomů, matici hodnot pro časové zlomy, matici svátků a výstupní proměnnou časového plánu. V případě, že se bude využívat i zadávání prázdnin (pomocí modulu Holiday), je nutné nadefinovat také matici prázdnin.

Pomocí parametru Mode/PlanningType modulů DayPlan / DayPlan2 lze pak nastavit, zda má být plánování úrovně nebo lineární (analogové).

Struktura matice časových zlomů a matice hodnot pro časové zlomy matice o rozměru $[n \times m]$, kde n musí být větší nebo rovno počtu zadávaných plánů pro jednotlivé dny a m je počet zlomů v daném časovém plánu. Sloupce matic určují jednotlivé body zlomu, tj. kolikrát se může plánovaná hodnota za den změnit (rozměr m). Řádky matic určují denní plány pro jednotlivé dny (rozměr n). V modulu DayPlan se každému řádku těchto matic přiřazuje specifický kód dne, podle kterého modul pozná (podle aktuálního data), který řádek má být aktivní. Kódy mohou určovat nejen jednotlivé dny v týdnu, ale i jejich možné kombinace.

Struktura matice svátků matice o rozměru $[3 \times n]$, kde n je počet svátků, které chceme zadat.

Struktura matice prázdnin matice o rozměru $[5 \times m]$ pro období prázdnin, kde m je počet období (úseků) prázdnin v roce.

Časové plány rozlišují zjednodušený a rozšířený režim. Ve zjednodušeném režimu musí být parametry modulu DayPlan nastaveny tak, aby bylo možné plánování uskutečnit pro každý den v týdnu plus svátek (matice časových zlomů a matice hodnot pro časové zlomy musí mít 8 řádků). V rozšířeném režimu je nutné využít v případech, kdy je požadavek na možnost editace různých skupin dnů v týdnu (pracovní dny, víkend, svátek) bez možnosti editace jednotlivých dnů. V tomto případě, je nutné, aby měly matice časových zlomů a matice hodnot pro časové zlomy tolik řádků, kolik je požadovaných skupin dnů. V tomto režimu se přidává navíc datová matice s názvy, které se budou zobrazovat při editaci.

Ovládání časových plánů se provádí přímo na PLC pomocí modulů TPlan, GPlan, Feasts a Holidays, které se programují jako obrazovky (Příloha VI). Modul TPlan slouží k textové editaci, kde se vybere požadovaný den, nadefinovaný časový zlom a přiřadí se mu žádaná hodnota a čas. Modul GPlan funguje na stejném principu, ale v grafické formě. Modul Feasts a Holidays jsou téměř stejné jako modul TPlan. U modulu Feasts se nastaví datum svátku a den, podle kterého se má plánovat. U modulu Holidays se nastavuje počáteční a koncové datum prázdnin a den, podle kterého se bude plánovat.

Programování časových plánů není těžké a zdouhavé. Program pro řízení od pondělí do neděle, o svátcích a prázdninách ve zjednodušeném režimu vypadá takto:

```
DayPlan 0x0000, 2, 0x001F, 0x00E0, 0x0000, 0x0000, 0x0000, 0x0000,  
        0x0000, 0x0000, DP_Svatky, DP_Casy, DP_Hodnoty, y4_zadana  
Holiday DP_Svatky[*,9], DP_Holiday
```

Ve zjednodušeném režimu jsou matice svátků a prázdnin prázdné. Rozměr matice svátků jsem zvolil 3x10 a rozměr matice prázdnin 5x6.

Matici časových zlomů jsem zvolil takto:

Tabulka 10. 1: Matice časových zlomů

1.1.1980 6:00	1.1.1980 10:00	1.1.1980 12:00	1.1.1980 15:00	1.1.1980 18:00	1.1.1980 20:00	1.1.1980 23:00
1.1.1980 6:00	1.1.1980 10:00	1.1.1980 12:00	1.1.1980 15:00	1.1.1980 18:00	1.1.1980 20:00	1.1.1980 23:00
1.1.1980 6:00	1.1.1980 10:00	1.1.1980 12:00	1.1.1980 15:00	1.1.1980 18:00	1.1.1980 20:00	1.1.1980 23:00
1.1.1980 6:00	1.1.1980 10:00	1.1.1980 12:00	1.1.1980 15:00	1.1.1980 18:00	1.1.1980 20:00	1.1.1980 23:00
1.1.1980 6:00	1.1.1980 10:00	1.1.1980 12:00	1.1.1980 15:00	1.1.1980 18:00	1.1.1980 20:00	1.1.1980 23:00
1.1.1980 8:00	1.1.1980 10:00	1.1.1980 12:00	1.1.1980 15:00	1.1.1980 18:00	1.1.1980 22:00	1.1.1980 23:00
1.1.1980 8:00	1.1.1980 10:00	1.1.1980 12:00	1.1.1980 15:00	1.1.1980 18:00	1.1.1980 22:00	1.1.1980 23:00
1.1.1980 8:00	1.1.1980 10:00	1.1.1980 12:00	1.1.1980 15:00	1.1.1980 18:00	1.1.1980 22:00	1.1.1980 23:00

Matici hodnot pro jednotlivé dny jsem zvolil takto:

Tabulka 10.2: Matice hodnot pro časové zlomy

20	18	18	20	22	22	18
20	18	18	20	22	22	18
20	18	18	20	22	22	18
20	18	18	20	22	22	18
20	18	18	20	22	22	18
20	22	22	22	22	20	18
20	22	22	22	22	20	18
20	20	20	20	22	20	18

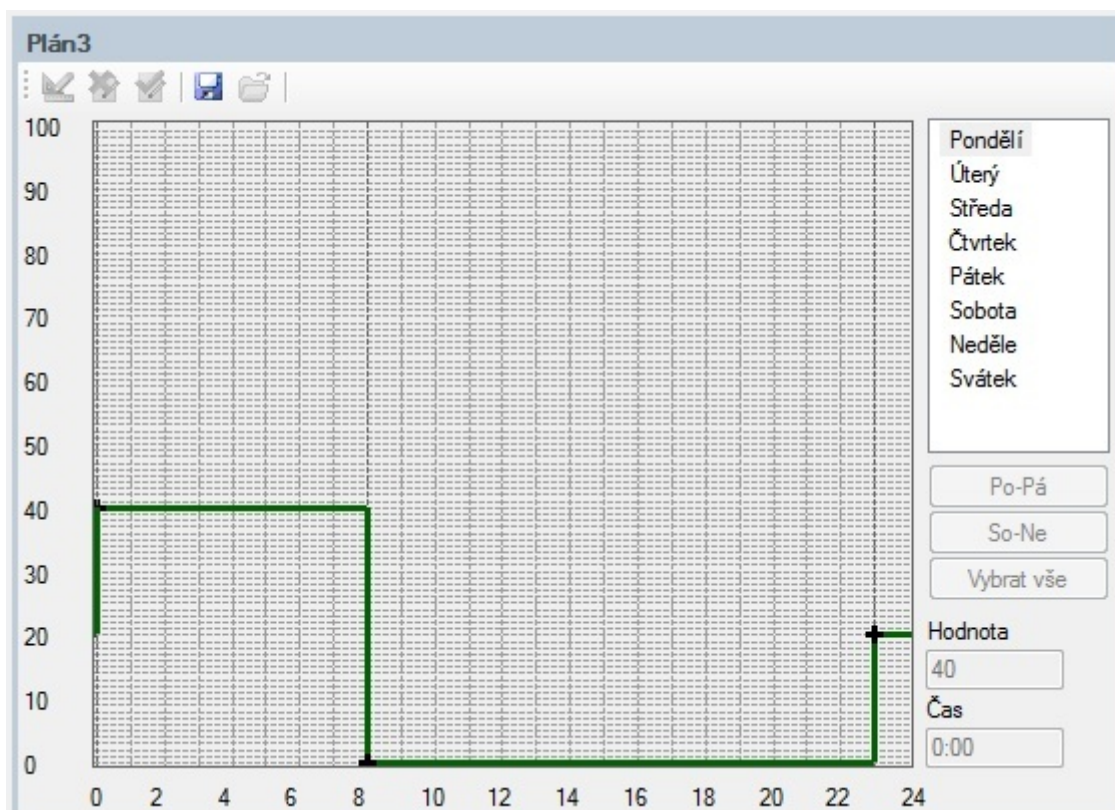
Z výše uvedených tabulek (Tabulek 10.1 a 10.2) lze vidět, že pro každý je vytvořeno 7 zlomů. První řádek matic odpovídá pondělí, předposlední neděli a poslední svátku.

Časové zlomy jsem volil tak, aby rovnoměrně pokryly celý den, ale zároveň tak, aby nebyly zvoleny v časech, kdy jsou absolutně nepoužitelné. Na dopoledne jsem zvolil 3 zlomy, a to tak, že první zlom je brzo ráno, aby když člověk vstává brzo ráno do práce, nevstával do vymrzlého pokoje. Další zlomy jsou nastaveny až déle a jenom dva, protože v těchto časech ve všedních dnech jsou téměř všichni mimo domov, a tyto zlomy slouží spíše jenom jako udržovací a také aby se změnila teplota z rána, kdy se místnost vytápěla. Zbýlé časové zlomy jsou voleny tak, aby pokryly odpoledne a večer. Ty jsem volil tak, že v průběhu odpoledne se většina lidí opět vrací domů, a mělo by se opět začít vytápět. Večerní nastavení je již plně na uživatelích, zda-li mají raději večer tepleji, nebo naopak chladněji. Poslední časový zlom jsem volil tak, aby sloužil k ukončení denního vytápěcího cyklu, tedy nastavením nízké teploty.

Co se týče teplot, zde jsem vycházel z úvahy, že běžná pokojová teplota je 20 °C. V časech, kdy by se uživatel nacházel doma, jsem volil zmíněnou pokojovou teplotu 20 °C. Když není doma, žádaná teplota není potřeba tolik vysoká, proto jsem zvolil 18 °C. Na večer, kdy se venkovní okolí ochlazuje, jsem zvolil teplotu vyšší, 22 °C.

S ohledem na to, že při aplikování tohoto programu na reálnou místnost nebo budovu, by se venkovní okolí ochlazovalo, a tím ovlivňovalo regulaci, jsem zavedl další časový plán, a to pro ventilátor č. 2, který se řídí spojitě. Ten je nastaven tak, aby se ve

večerních a ranních hodinách spínal nejprve na 20 % a poté na 40 % výkonu, a tím simuloval zmíněné venkovní ochlazení. Nastavení lze vidět na obrázku 10.1.

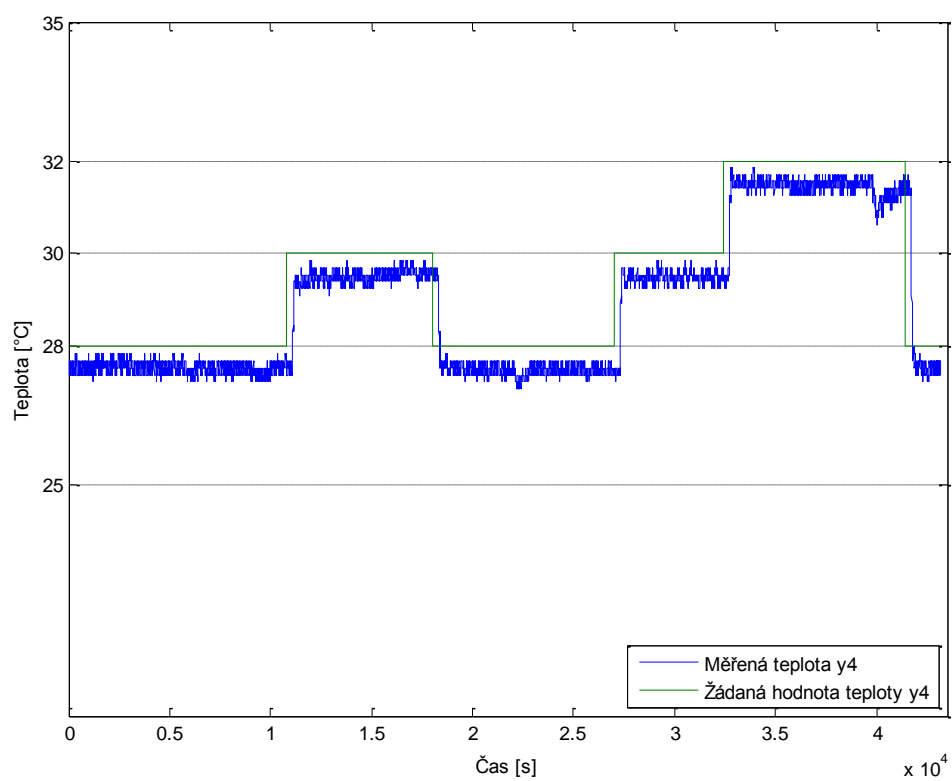


Obrázek 10.1: Nastavení hodnot časového plánu pro ventilátor č. 2

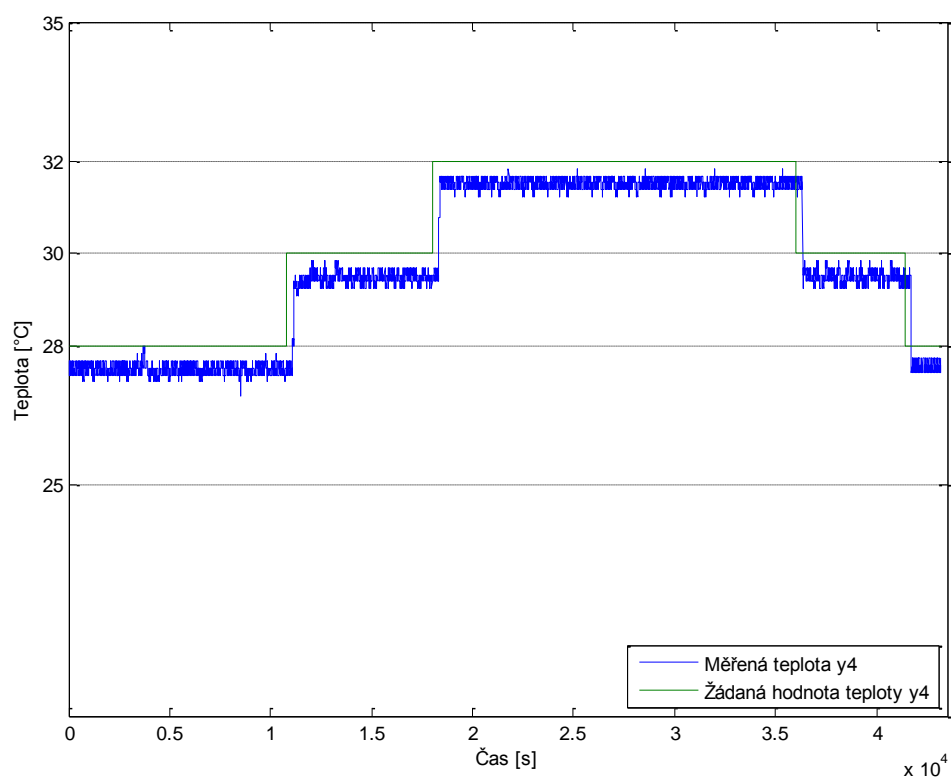
Pro ověření funkce časových plánů jsem udělal dvě změny. První změnou bylo jiné nastavení teplot. To jsem musel udělat proto, že model byl umístěn v učebně, kde byla běžně teplota přibližně 22 °C, a měření by bylo proto tímto ovlivněno. Všechny žádané hodnoty teploty jsem tedy zvýšil o 10 °C.

Další změnou byla změna doby cyklu. Výše uvedené nastavení časových zlomů je 24 hodinové. Měření jsem provedl na 12ti hodinovém cyklu, a to tak, že jsem doby mezi zlomy zmenšil na polovinu.

Měření jsem provedl na nastavení pro všední den a víkend. Výsledné průběhy jsou vidět na obrázku 10.2 a 10.3.



Obrázek 10.2: Průběh teploty y_4 při měření časových plánů



Obrázek 10.3: Průběh teploty y_4 při měření víkendového režimu

11. Kompletní program v DetStudios

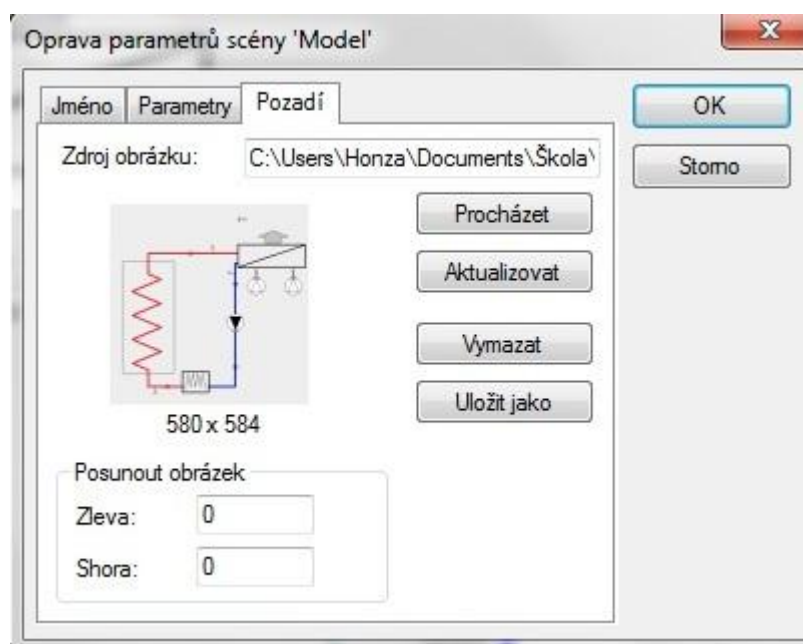
Kompletní program pro řízení modelu soustavy se skládá z již zmiňovaných částí, které jsou umístěny v jednotlivých procesech. Výpis použitých proměnných a aliasů je v Příloze V. Program obsahuje dva procesy pro ovládání prvků modelu soustavy a načítání a archivaci teplot (viz. kapitola 6.1). Další proces obsahuje část programu s časovými plány, které jsou podrobněji popsány v kapitole 10. Poslední proces řídí regulaci, ve kterém je obsažen PID regulátor, jehož kód je zmíněn v kapitole 8.1. Poslední částí programu jsou obrazovky, které byly naprogramovány pro potřeby řešení této úlohy. Všechny obrazovky jsou zobrazeny v příloze VI i s popisem.

12. Vizualizace

Pro správu dat a řízení modelu soustavy pomocí PLC jsem vytvořil jednoduchou vizualizaci (Příloha VI), která se skládá ze 4 scén. První scéna je věnována obecnému přehledu o modelu. Je zde na pozadí vloženo jednoduché schéma modelu soustavy. U jednotlivých prvků schématu jsou vloženy prvky zobrazující hodnoty teplotních čidel a hodnoty jednotlivých výkonů prvků. Na této základní scéně jsou vloženy i informace o nastavení regulátoru a žádané teplotě. Na scéně je i prvek PCArchiv, kde jsou zobrazeny průběhy teplot.

Další scéna obsahuje pouze prvek pro ovládání časových plánů. Třetí scéna obsahuje prvek pro zobrazení provozního deníku PLC. V poslední scéně je umístěn archiv PLC teplot, který je obsažen v programu.

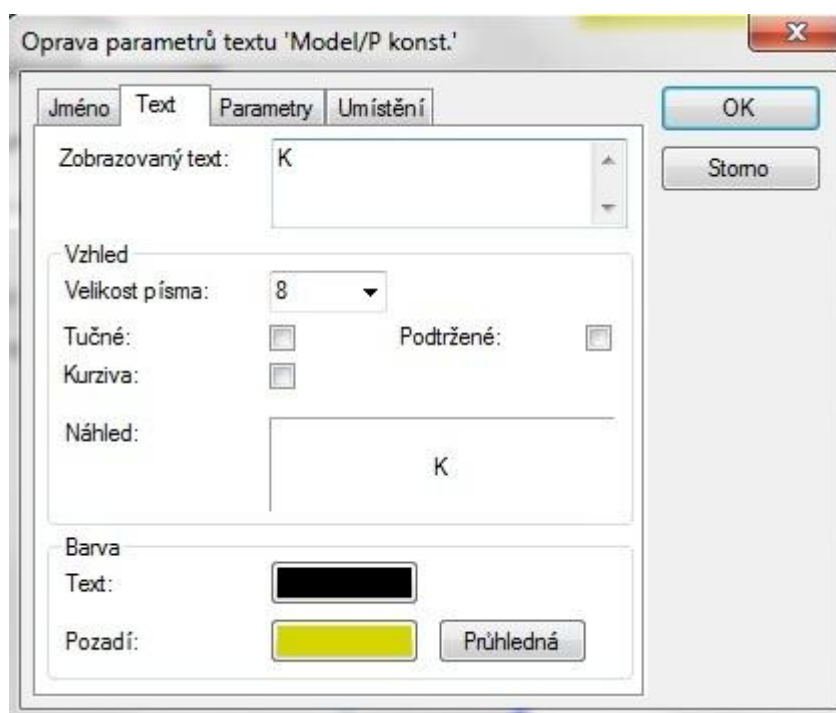
Na níže uvedených obrázcích je zobrazeno konkrétní nastavení jednotlivých prvků. Pro prvky, které jsou obsaženy ve vizualizaci vícekrát, je zde uvedeno jen pro jeden prvek, aby byly vidět možnosti a způsob nastavení.



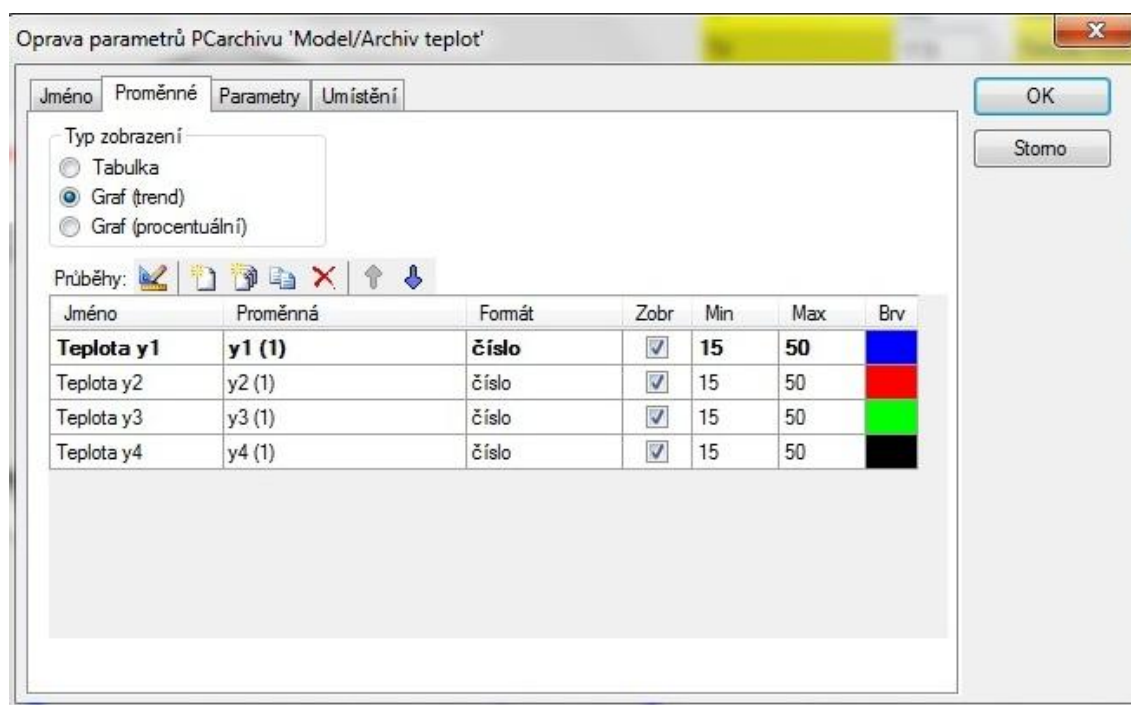
Obrázek 12.1: Nastavení pozadí scény



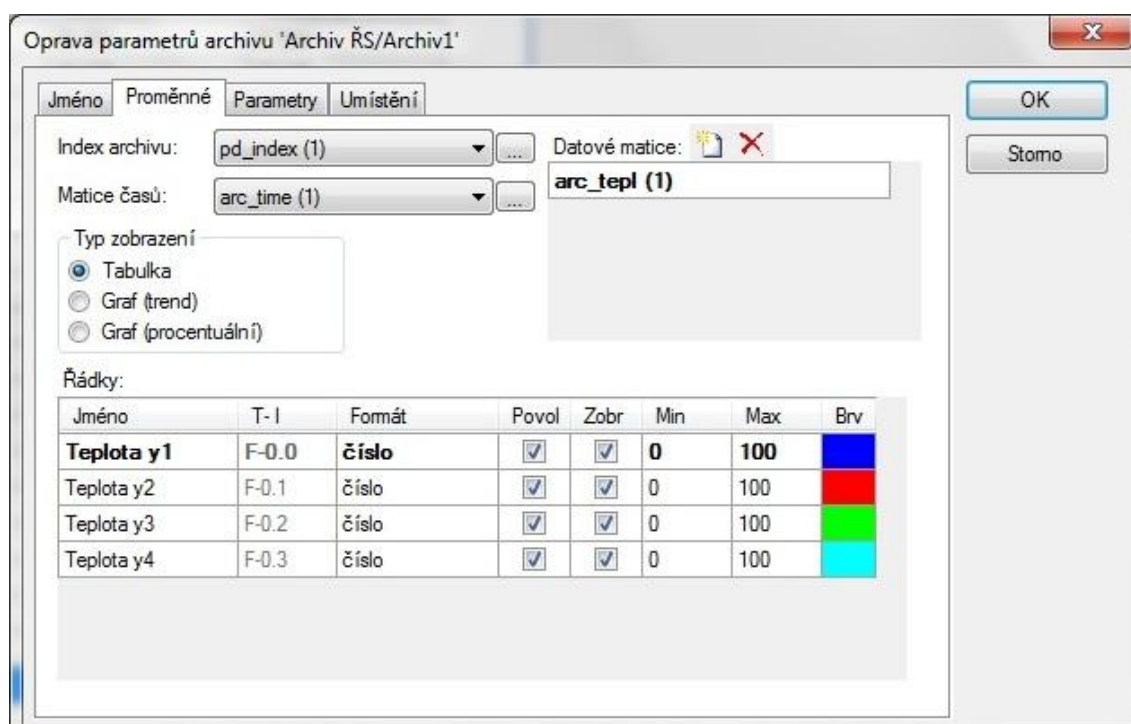
Obrázek 12.2: Nastavení prvku Proměnná



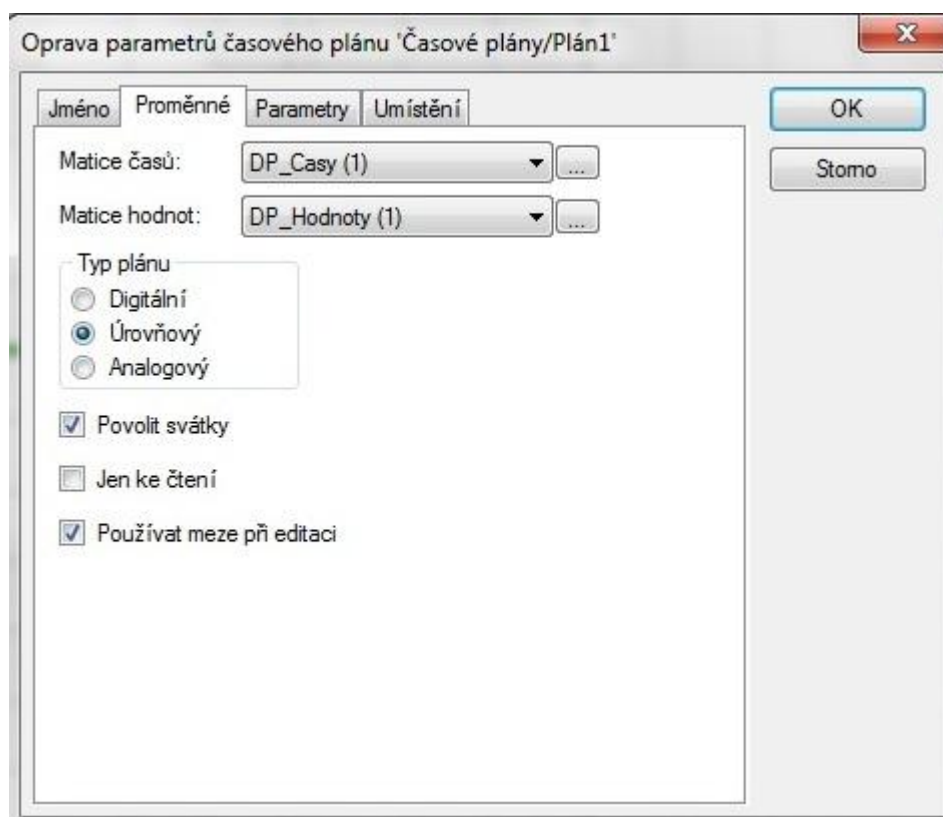
Obrázek 12.3: Nastavení prvku Text



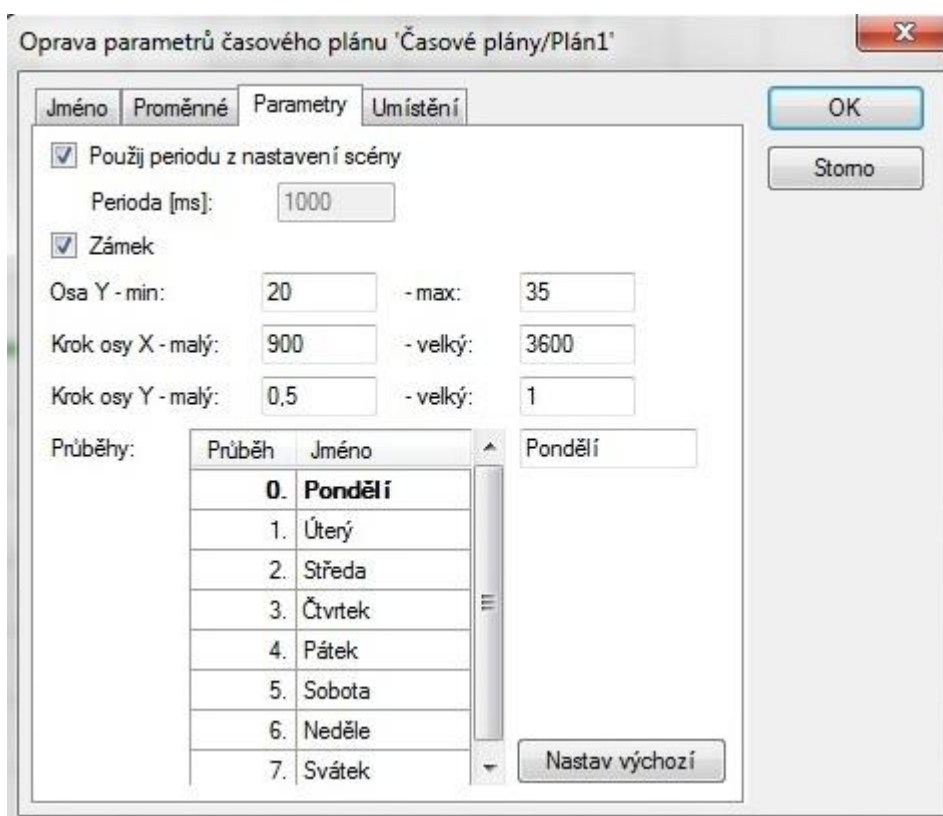
Obrázek 12.4: Nastavení prvku PCArchiv



Obrázek 12.5: Nastavení prvku Archiv



Obrázek 12.6: Nastavení proměnných prvku Časový plán



Obrázek 12.7: Nastavení parametrů prvku Časový plán

Závěr

Řídicí systém AMiNi4DS se díky svým vlastnostem a velkým množstvím předdefinovaných funkcí nejlépe hodí pro řízení různých měření a regulací, ovládání inteligentních domů, ovládání menších strojů, jako jsou například různé kotle, bojlerů atd. Mezi jeho velké přednosti patří, že je to produkt českého výrobce (AMiT), takže je dostupný a všem srozumitelný, je jednoduchý, kompaktní, má širší možnosti použití a je navržen tak, aby nebyl unikátní, tedy aby k jeho obsluze nebylo zapotřebí různých speciálních zařízení. Například různé redukce.

Vývojové prostředí, DetStudio, má mé osobní hodnocení velmi vysoké. Velikou předností je podpora českého jazyka, takže vše je srozumitelné, což velmi usnadňuje práci a hledání nejen v nápovědě, kde jsou uvedeny popisy jednotlivých modulů, ze kterých se skládá program. Další výhodou jsou již předem předdefinované moduly, funkční bloky a mnoho dalších potřebných a užitečných funkcí, které není třeba složitě programovat a vymýšlet. Mezi užitečné, a jistě velmi používané, patří možnosti tvorby pracovních deníků, různých archivací dat a archivů, tvorby vlastních obrazovek atd. Drobným nedostatkem je menší možnost práce s proměnnými, hlavně v jejich porovnávání a v jejich možnosti vkládání je jako různé podmínky.

Servisní nástroj ViewDet má též mé vysoké hodnocení. Je to velmi jednoduchý, přehledný a intuitivní program. Práce s ním je velmi jednoduchá díky tomu, že opět jako v Detstudiu, spočívá ve vkládání scénických prvků na scénu. Editace těchto prvků je velmi lehká. Velikou výhodou je snadný export dat, který lze pak snadno zpracovat v dalších programech.

Co se týče jednotlivých bodů zadání, tak ty se mi podařilo splnit všechny. Jejich stručné řešení a výsledky jsou popsány v následujících odstavcích.

První část identifikace modelu, nastavení optimálního výkonu čerpadla, nebyla složitá, ale byla zdoluhavá, protože si vyžádala sérii delších měření. Nastavení výkonu čerpadla jsem odhadoval na 75 %, což se následně potvrdilo na základě provedených měření.

Druhá část identifikace spočívala ve změření přechodové charakteristiky pro teplotu y4. Zde byl postup lehčí, protože byl hodně zjednodušen programem Matlab, který provedl identifikaci modelu sám, na základě mnou zvolených parametrů a naměřených dat přechodové charakteristiky. Aproximované systémy vyšly s přesností vyšší než 95 %.

Pro návrh a porovnání parametrů pro PID regulátor jsem využil několik metod. Metodu Zieglera a Nicholse, Kuhnovu metodu, nastavení pomocí bloku PID v programu Matlab a metodu Chien, Hronese a Reswicka. Podle očekávání použití metody Zieglera a Nicholse je v tomto případě nevhodné. Jako nejlepší metoda vyšla metoda Chien, Hronese a Reswicka, která je přímo navržena tak, že nejlepšími výsledky dosahuje pro systémy prvního s dopravním zpožděním. Nejlehčí metoda je použití bloku PID v programu Matlab. Zde stačí jednoduché simulační schéma a Matlab provede celý návrh sám. Tento způsob, ale není zcela nejlepší, řekl bych, že je spíše univerzální.

Model jsem reguloval pomocí PID regulátoru. Zde jsem narazil na dva zásadní problémy. Prvním byla špatná funkce implementovaného PID regulátoru v programu DetStudio. Potíž spočívala v tom, že regulátor měl na výstupu akční zásah 0 nebo 100 %. Po několikanásobné změně a především pečlivé kontrole jeho nastavení a parametrů, se mi tento problém nepodařilo odstranit. Proto jsem zvolil možnost naprogramovat

vlastní PID regulátor. U tohoto způsobu řešení jsem narazil na druhou komplikaci, která spočívala ve správné volbě výpočtu akčního zásahu. Zde jsem vyzkoušel více než deset výpočtů. Vesměs se potvrdovalo, že čím jednodušší výpočet jsem zvolil, tím jsem dosáhl lepších výsledků. Výsledný průběh regulované veličiny, teploty y_4 , je téměř bez překmitu a ustálení je přibližně $0,5\text{ }^{\circ}\text{C}$, což považuju za tolerovanou mez. Regulátor ovšem má pomalejší odezvu na chybu, zde je ustálení přibližně po 15ti minutách, což není úplně ideální, ale lepších výsledků jsem nedosáhl.

Jako druhou variantu regulace modelu, jsem zkoušel Smithův prediktor, který je vhodnější pro regulaci systémů s dopravním zpožděním. Protože prediktor není v DetStudiosu implementován, tak jsem ho musel naprogramovat. Naprogramovat prediktor bylo mnohem náročnější než tvorba samotného PID regulátoru a proto se ve výpočtu akčního zásahu vyskytla chyba, kterou se mi nepodařilo odstranit, a proto regulace pomocí prediktorku není přesná a plně použitelná.

Naprogramování časových plánů bylo díky rozsáhlé podpoře relativně snadné. Na uživateli pak už jenom záleží, kolik zvolí časových zlomů pro dny a jaké si zvolí hodnoty v definovaných zlomech. Toto nastavení se provádí snadno, pomocí proměnné typu matice, kde počet sloupců je roven počtu zlomů. V DetStudiosu lze pouze nastavit pevně zvolené časové zlomy a to tak, že se vyplní zmiňovaná matice pevně zvolenými časy. Jejich úprava je možná až přímo z PLC pomocí obrazovek, které slouží k editaci časových plánů, nebo pomocí programu ViewDet, kde lze pomocí jednoduchého scénického prvku provést kompletní editaci časového plánu.

Pro tuto úlohu jsem vytvořil dva časové plány. První časový plán slouží k ovládání ventilátoru č. 2, kterým se simuluje venkovní změna teploty, především v noci, kdy jsem v časovém plánu nastavil jeho spínání. Druhý byl pro ovládání modelu, regulaci teploty y_4 . Zde jsem vytvořil časový plán pro všední dny a víkend. Časové plány jsem ověřil pomocí měření a výsledky byly uspokojivé, protože změny žádaných hodnot a regulace samotná vyšla dle očekávání, a to jak pro všední dny, tak i pro víkendový režim.

Seznam použité literatury

- [1] PODOLÁK, Stanislav. *AMiNi4DS, Návod na obsluhu* [online]. 2009 [cit. 2012-05-10]. AMiT.
URL: <http://www.amit.cz/docs/cz/amini/amini4ds_g_cz_100.pdf>
- [2] ŘÍHA, Zbyněk. *DetStudio, Návod na obsluhu* [online]. 2009 [cit. 2012-05-10]. AMiT.
URL: <http://www.amit.cz/docs/cz/sw/detstudio_g_cz_101.pdf>
- [3] *AMiNi4DS, katalogový list* [online]. 2009 [cit. 2012-05-10]. AMiT.
URL: <http://www.amit.cz/docs/cz/amini/amini4ds_d_cz_100.pdf>
- [4] ŘÍHA, Zbyněk. *ViewDet, Návod na obsluhu* [online]. 2009 [cit. 2012-05-10]. AMiT.
URL: <http://www.amit.cz/docs/cz/sw/viewdet_g_cz_100.pdf>
- [5] HLAVA, Jaroslav. *Prostředky automatického řízení II* [online]. 2000 [cit. 2012-05-10]. AMiT.
URL: <http://www.fm.tul.cz/~jaroslav.hlava/par/Skripta_PAR.pdf>
- [6] ŘÍHA, Zbyněk. *Použití časových plánů* [online]. 2010 [cit. 2012-05-10]. AMiT.
URL: <http://www.amit.cz/support/cz/aplikacni_poznamky/ap0031_cz_02.pdf>
- [7] ŘÍHA, Zbyněk. *Nastavení PID* [online]. 2010 [cit. 2012-05-10]. AMiT.
URL: <http://www.amit.cz/support/cz/aplikacni_poznamky/ap0042_cz_01.pdf>
- [8] ŘÍHA, Zbyněk. KUČERA, Jan. *Vizualizace* [online]. 2010 [cit. 2012-05-10]. AMiT.
URL: <http://www.amit.cz/support/cz/aplikacni_poznamky/ap0012_cz_02.pdf>
- [9] *Výukový model tepelné soustavy s dopravním zpožděním*. Univerzita Tomáše Bati ve Zlíně, 2006.
- [10] BERAN, Jan. *Řízení tepelné soustavy pomocí PLC*. Technická Univerzita v Liberci, 2011.
- [11] *Diskrétní regulátory – modifikace spojitých regulátorů* [online]. [cit. 2012-05-10].
URL: <http://195.178.89.122/CAAC_PHP/CAAC/cesky/synteza/d_reg/d_reg.php>
- [12] *REGULACE, část 2: Číslicová regulace* [online]. 2007 [cit. 2012-05-10]. Ústav počítačové a řídicí techniky VŠCHT Praha.
URL: <http://moodle.vscht.cz/pluginfile.php/4514/mod_resource/content/0/PRS-eg_2.pdf>
- [13] *Řízení systému s dopravním zpožděním* [online]. [cit. 2012-05-10]. Univerzita Tomáše Bati ve Zlíně. URL: <<http://rtp.webzdarma.cz/rizeni15.php>>

Příloha I – Katalogový list PLC AMiNi4DS



Compact Control Systems

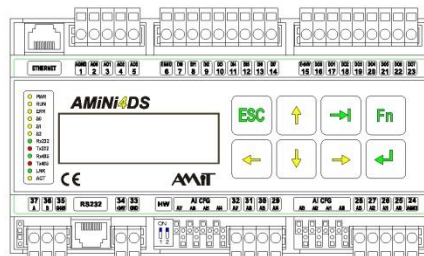
Revision: X/09

amini4ds_en_100

AMiNi4DS

Compact Control System with Display

- 8 × digital output 24 V / 0.3 A DC
- 8 × digital input 24 V AC/DC
- 8 × analogue input U/I/Ni1000/Pt1000
- 4 × analogue output 0 to 10 V
- RS232, RS485 with GS, Ethernet
- 35 mm DIN rail mounting
- LCD display 122 × 32 pixels, keyboard



TECHNICAL DATA

CPU	ST10F269
FLASH memory	256 + 1024 KB
RAM memory (battery back-up)	1024 KB
EEPROM memory	2 KB
Back-up battery lifetime (RAM + RTC)	Panasonic lithium battery, 5 years
Inputs	8 × DI + 8 × AI
Digital inputs	24 V AC/DC
Analogue inputs	0 to 5 V / 0 to 10 V / 0 to 20 mA / Ni1000 / Pt1000
Galvanic separation of digital inputs	Yes *)
Galvanic separation of analogue inputs	No
Outputs	8 × DO + 4 × AO
Digital outputs	24 V / 0.3 A DC
Analogue outputs	0 to 10 V (max. 10 mA)
Galvanic separation of digital outputs	Yes *)
Galvanic separation of analogue outputs	No
Communication	
Serial communication interface	1 × RS232 (RJ45), in accordance with EIA-561 1 × RS485 (WAGO connectors)
Ethernet interface	1 × 10 Mbps, RJ45, in accordance with IEEE802.3
Galvanic separation of RS232	No
Galvanic separation of RS485	Yes *)
Power supply	24 V DC ±20 %
Power consumption (without outputs)	Max. 200 mA at 24 V DC
Others	
Display	Graphical, 122 × 32 pixels
Keyboard	8 keys
Signal connection	WAGO 231 cage clamp connectors
Cover protection rate	IP20
Operating temperature	0 to 50 °C
Max. ambient humidity	< 95 % non-condensing
Mounting	35 mm DIN rail
Weight	500 g
Dimensions (w × h × d)	160 × 98 × 74 mm

*) Insulation strength 300 V AC / 1 minute, galvanic separation may not be used for safe and unsafe parts separation.

ORDERING INFORMATION

AMiNi4DS	Compact control system, full connector set, user's manual, warranty card
KABEL 232RP	RS232 to PC connecting cable

ISO 9001

www.amit.cz

AMIT, spol. s r.o., Naskové 3/1100, 150 00 PRAGUE, Czech Republic
Tel.: +420 222 780 100, +420 222 781 516, Fax: +420 222 782 297
e-mail: amit@amit.cz

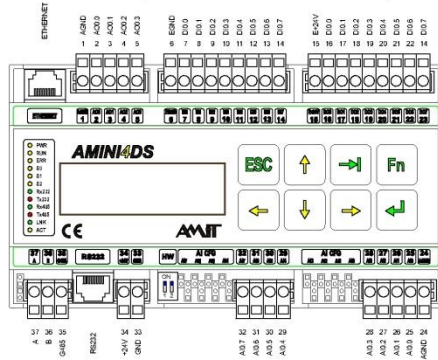
Technical support: support@amit.cz
Tel.: +420 549 210 276, +420 549 210 403
www.amit.cz/support



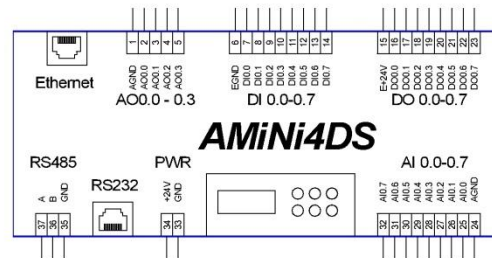
TERMINALS IDENTIFICATION

Analogue outputs	AO	4 × 10-bit analogue output without galvanic separation	1 .. 5
Digital inputs	DI	8 × galvanically separated input	6 .. 14
Digital outputs	DO	8 × galvanically separated output	15 .. 23
Analogue inputs	AI	8 × 10-bit analogue input without galvanic separation	24 .. 32
Power supply	PWR	power supply 24 V DC	33 .. 34
Communication	RS485		35 .. 37

TERMINAL LOCATION

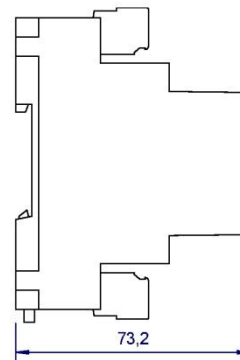
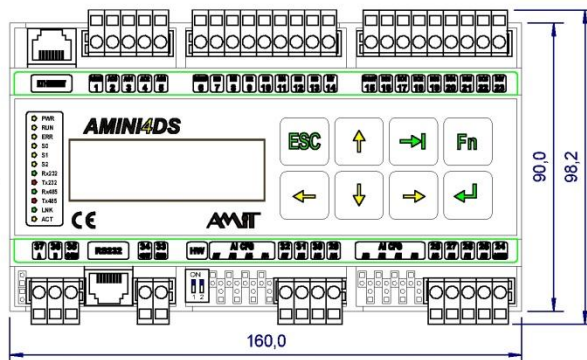


RECOMMENDED DIAGRAM SYMBOL

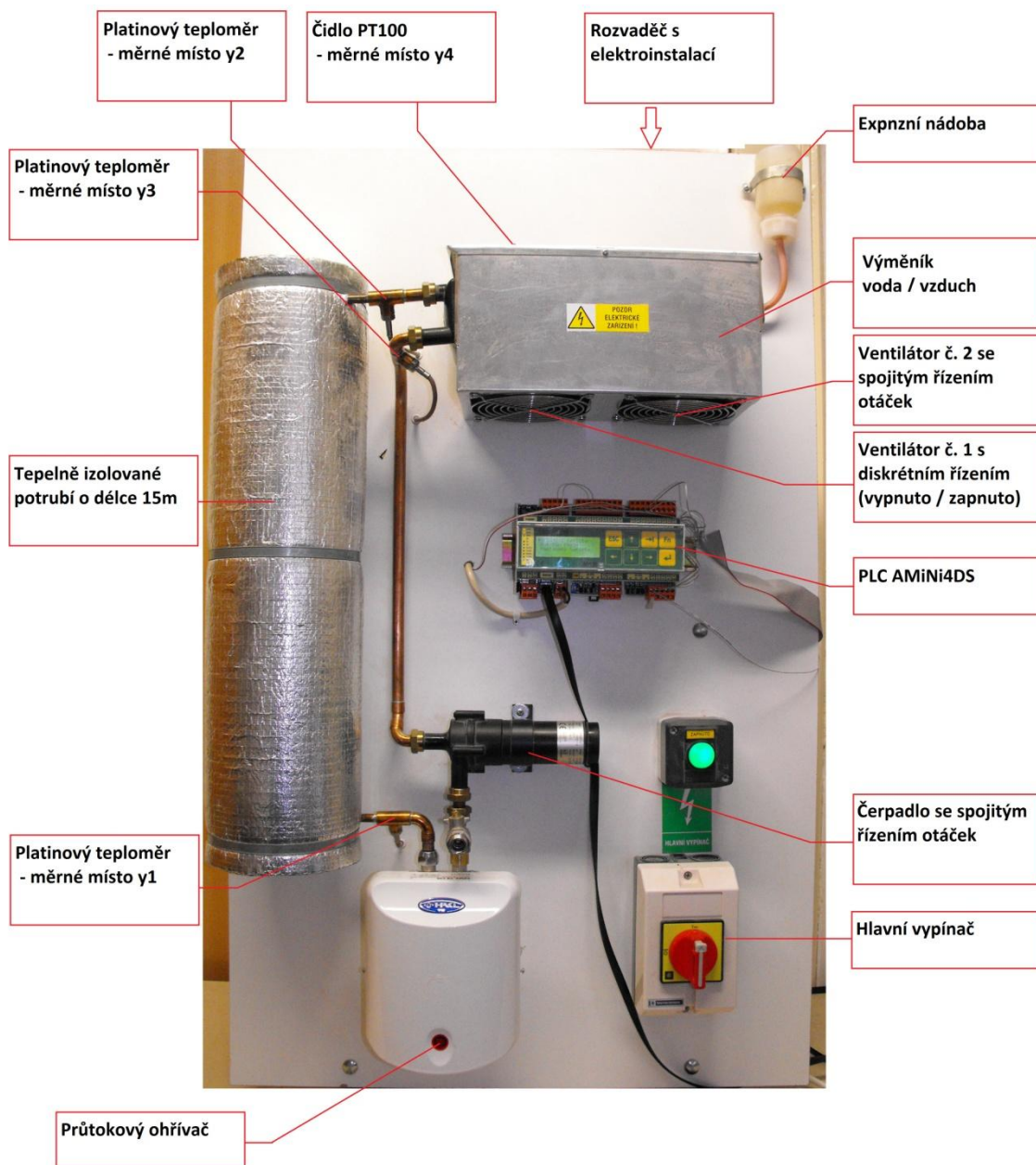


Note: GND, AGND and RS232-GND terminals are internally connected

PHYSICAL DIMENSIONS



Příloha II – Model tepelné soustavy s popisem



Příloha III – Skript pro výpočet doby náběhu a průtahu

```
jm=[10840.41135 522.207 1];
sys=tf(36.41,jm); %vytvoření funkce přenosu systému
n=100000; %počet vzorků
tmax=3000; %maximální čas
t=linspace(0,tmax,n); %vytvoření vektoru
T=tmax/(n-1);
y=step(sys,t); %vytvoření skokové funkce
dy=diff(y)/T; %derivace
[k i]=max(dy); %souřadnice maxima derivace
ti=t(i);
tecna=k*(t-ti)+y(i); %predpis primky
plot(t,y,t,tecna) %vykreslení do grafu
axis([0 3000 -10 50]) %nastavení os
grid on

Tn1= 0;
Tn= 0;
Tu= 0;
a=1;
for i=1:1:(n-1) %výpočet doby náběhu a průtahu
    if tecna(i)>= 0 & a==1
        Tu=t(i);a= 0;
    end
    if tecna(i)>=y(n)
        Tn1=t(i);break;
    end
end
Tn=Tn1-Tu;
```

Příloha IV – Smithův prediktor

```
AnIn #AI00_3, y4, 10.000, 0.6, 6.9, 0.000, 100.000
```

```
If @automaticky
```

```
  Let err3 = err2
```

```
  Let err2 = err1
```

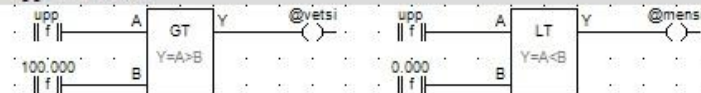
```
  Let y4_zadana_1 = y4_zadana - y4_r
```

```
  Let err1 = y4_zadana_1 - y
```

```
  Let up = (Td + (Td/Ti) + Td*K)*err1 - (Td + 2*(Td*K))*err2 + (Td*K)*err3
```

```
  Let u = u + up
```

```
  Let upp = 100*u
```



```
If @vetsi
```

```
  Let upp = 100
```

```
EndIf
```

```
If @mensi
```

```
  Let upp = 0
```

```
EndIf
```

```
AvgVar upp, up_p, 10
```

```
Let topeni = up_p
```

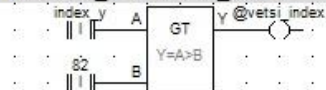
```
Let y = 0.076345*uppp + 0.9979*yp
```

```
SyncMark 1, 1, 0, 0, 1, @archivuj_y, NONE
```

```
SyncArch y, 0, arc_y[0,*], arc_y_time, @archivuj_y, @index, 3, arc_index_y, 0x0001
```

```
If @index
```

```
  Let index_y = index_y + 1
```



```
If @vetsi_index
```

```
  Let index_y = 0
```

```
EndIf
```

```
EndIf
```

```
SyncMark 1, 83, 0, 0, 1, @tick, NONE
```

```
If @tick
```

```
  Let @tick_1 = true
```

```
EndIf
```

```
If @tick_1
```

```
  Let y4_arc = arc_y[0,index_y]
```

```
Else
```

```
  Let y4_arc = 0
```

```
EndIf
```

```
Let y4_r = y4 - y4_arc
```

```
Let uppp = upp/100
```

```
Let yp = y
```

```
EndIf
```

Příloha V – Proměnné a Aliasy

Proměnná	WID	Typ	Stanice	Init	Popis
ventilator	1004	F	1		Řídicí napětí ventilátor2
cerpadlo	1000	F	1	75	Řídicí napětí čerpadlo
y1	1001	F	1		cidlo y1
y2	1003	F	1		cidlo y2
y3	1002	F	1		cidlo y3
y4	1016	F	1		čidlo y4
y4_zadana	1020	F	1		žádaná teplota y4
ovladani	1014	I	1		ovládání topení a ventilátoru1
pomocna	1010	I	1		pomocná proměnná
arc_index	1012	I	1		index archivu
arc_tepl	1013	MF[4,1000]	1		archiv teplot
arc_time	1015	ML[1,1000]	1		matice casu
pd_index	1900	I	1		index provozního deníku
pd_buffer	1901	MI[1,808]	1		rozměr pro 100 hlášení
topeni	1011	F	1		ovládání topení pomocí pwm
perioda_pwm	1017	F	1	4	perioda PWM modulace
vent	1019	F	1	100	ovládání ventilátoru1 pomocí pwm
K	1023	F	1	17.5	proporciální konstanta
Ti	1024	F	1	540	integrační konstanta
Td	1025	F	1	0.11	derivační konstanta
u	1027	F	1		akční zásah
up	1028	F	1		přírůstek akčního zásahu
up_p	1026	F	1		průměrovaná hodnota akčního zásahu
upp	1008	F	1		pomocná proměnná akčního zásahu
pomocna_1	1029	I	1		pomocná proměnná
err1	1033	F	1		k
err2	1034	F	1		k-1
err3	1035	F	1		k-2
DP_Svatky	1018	MI[3,10]	1		Matice svátků pro časový plán
DP_Out	1009	F	1		Výstup časového plánu
DP_Holiday	1007	MI[5,6]	1		Matice prázdnin pro modul Holiday
DP_Hodnoty_c	1022	ML[8,3]	1		Hodnoty pro plán chyby
DP_Hodnoty	1006	MF[8,7]	1		Hodnoty pro plán
DP_Casy_ch	1021	ML[8,3]	1		Matice časů plánu chyby
DP_Casy	1005	ML[8,7]	1		Matice časů plánu
Alias	Proměnná	Bit	Popis		
@topeni	ovladani	1	ovládání topení		
@vent	ovladani	0	ovládání ventilátoru1		
@autimaticky	pomocna	0	volba automatického nebo manuálního režimu		
@archivuj	pomocna	7	bit pro archivaci		
@vetsi	pomocna_1	0	bit porovnání		
@mensi	pomocna_1	1	bit porovnání		

Příloha VI – Přehled vytvořených obrazovek

```
►Aktualní teploty
Aut-Man režim
Nastavení teploty
```

Základní Menu obrazovka

```
cidlo y1      0,0 °C
cidlo y2      0,0 °C
cidlo y3      0,0 °C
cidlo y4      0,0 °C
```

Obrazovka pro zobrazení jednotlivých hodnot teplot

```
Regulace
Zadana tep. ► 0 °C
<Ent>edit <Esc>zpet
```

Obrazovka pro zadání žádané teploty

```
Ovladani ventilatoru
Vent 1 ► 0 %
Vent 2 ► 0 %
<Ent>edit <Esc>zpet
```

Obrazovka pro ovládání jednotlivých ventilátorů

```
►Graficky
Textově
Svátky
Prázdniny
```

Menu obrazovka pro časové plány

```
12:45 22.5
Po-Pá 3
```

Obrazovka pro textové ovládání časových plánů

```
Ovladani cernpadla
Topeni ► 0 %
Cernpadlo ► 0 %
<Ent>edit <Esc>zpet
```

Obrazovka pro ovládání čerpadla a průtokového ohřevače

```
Aut/Man režim
Aut. režim ⚡OFF
<Ent>edit <Esc>zpet
```

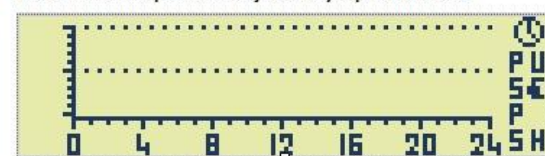
Obrazovka pro volbu ručního, nebo automatického režimu

```
0 00:00:01
```

Obrazovka pro zobrazení provozního deníku

```
►Ventilátory
Čerpadlo a topení
```

Menu obrazovka pro ovládání jednotlivých prvků modelu



Obrazovka pro grafické ovládání časových plánů

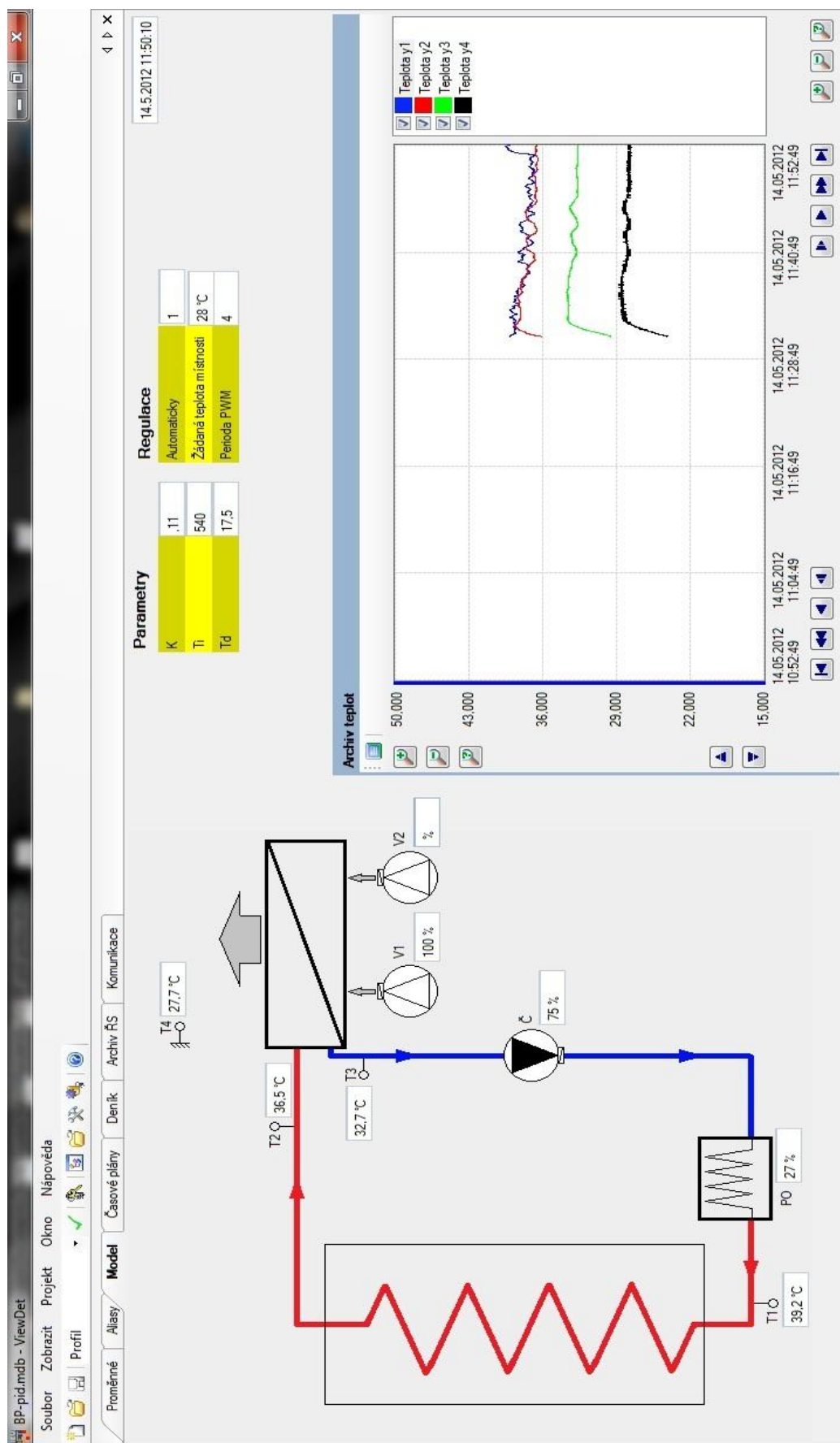
```
Date1 Index
Value
```

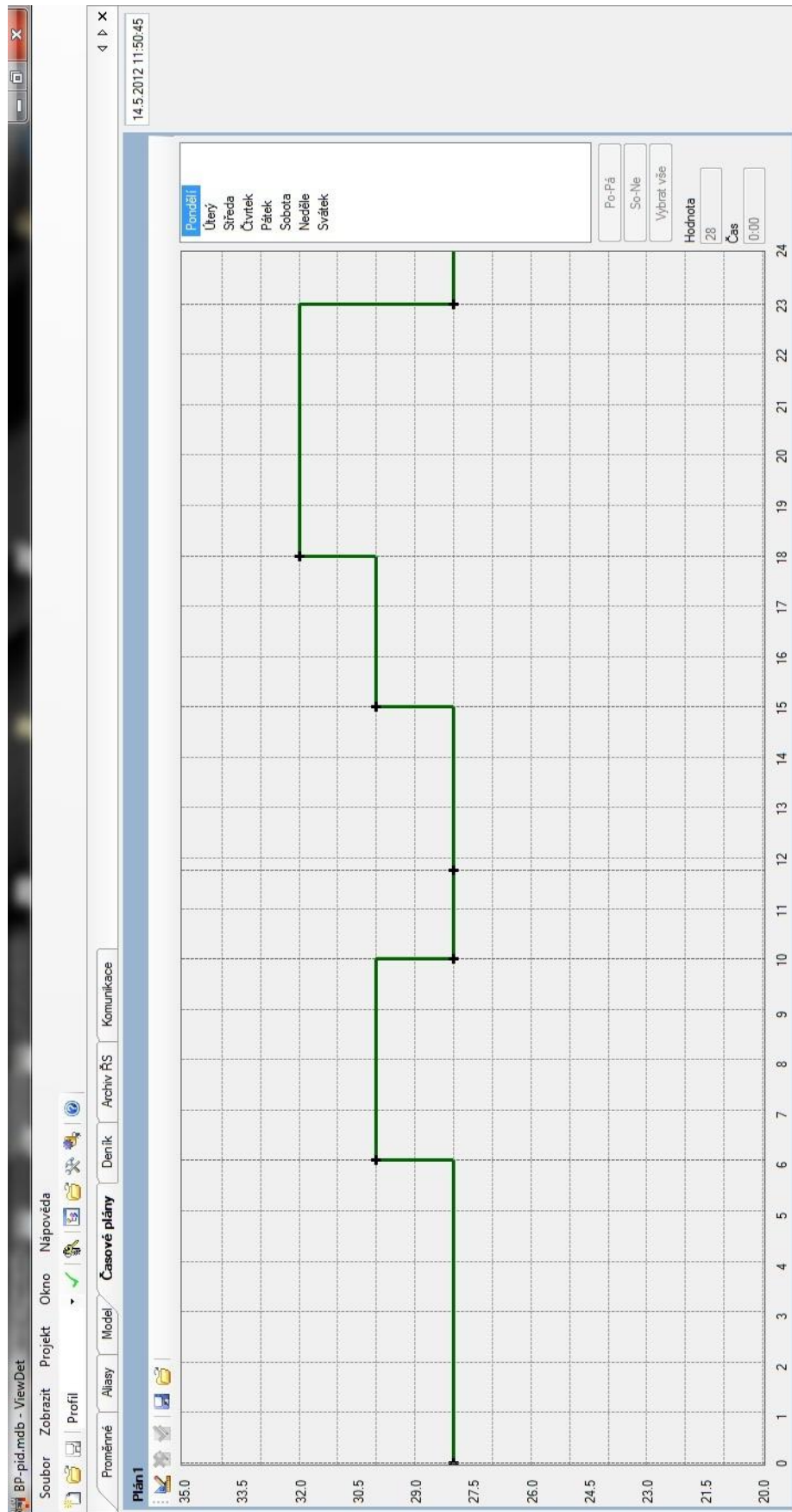
Obrazovka pro nastavení svátků

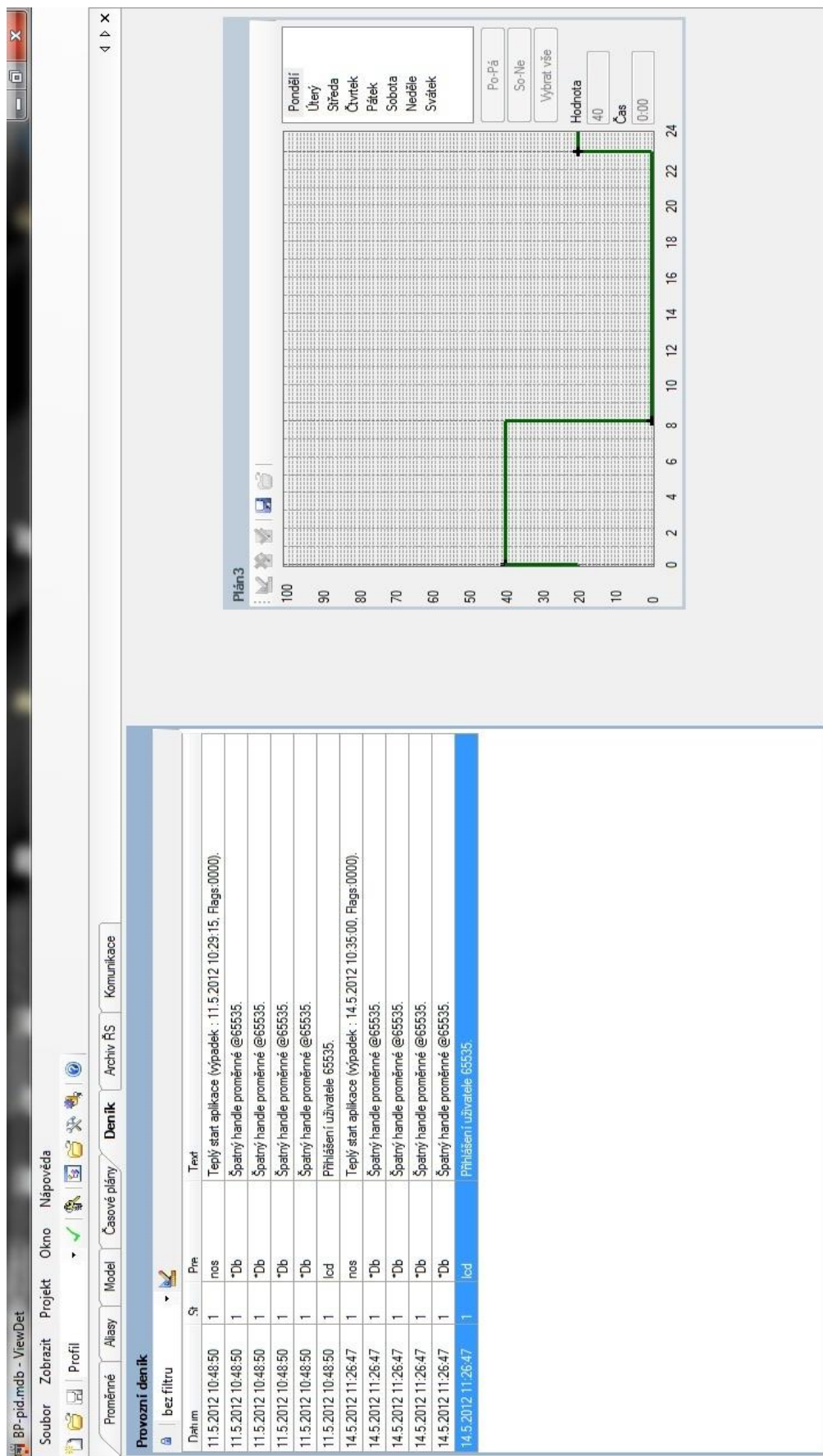
```
Date1 Date2
Value Index
```

Obrazovka pro nastavení prázdnin

Příloha VII – Jednotlivé scény vizualizace







BP-pid.mdb - ViewDet

SouborZobrazitProjektOknoNápověda

</

BP-pid.mdb - ViewDet

Soubor Zobrazit Projekt Okno Napověda

Profil

Proměnné Aliasy Model Časové plány Deník Archiv RS **Komunikace**

Stanice	Hodnota
AMN	14.5.2012 11:51:40
(1)	pasivní
	AMN4(D)S V1.00 - NOS166 V3.57 - DbNET V1.51
	aplikace AMT spol. s r.o. Praha - AMN4(D)S V1.00 - NOS166 V3.57 - DbNET V1.51
	aplikace Bakařská práce - TUL Liberec - BP-pid-pokus.11.05.12 V1.6.17